Universidad Autónoma de Madrid
Escuela politécnica superior

Doble Grado en Ingeniería Informática y Matemáticas

Trabajo Fin de Grado

ANÁLISIS E IMPLEMENTACIÓN DE DIFERENTES MEDIDAS DE SIMILITUD PARA UN ALGORITMO GLOBAL DE SELECCIÓN DE VARIABLES

Autor: Sara Dorado Alfaro
Tutor: Irene Rodríguez Luján
Ponente: José Ramón Dorronsoro Ibero

MAYO 2017

ANÁLISIS E IMPLEMENTACIÓN DE DIFERENTES MEDIDAS DE SIMILITUD PARA UN ALGORITMO GLOBAL DE SELECCIÓN DE VARIABLES

Autor: Sara Dorado Alfaro
Tutor: Irene Rodríguez Luján
Ponente: José Ramón Dorronsoro Ibero

Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
MAYO 2017

Resumen

Resumen

La selección de variables es un paradigma dentro del Aprendizaje Automático que trata de *elegir o seleccionar* las variables más importantes o relevantes en un problema de reconocimiento de patrones. Hoy en día tiene especial interés debido a la aparición de problemas de alta dimensionalidad, en los que puede haber variables que no aportan información útil o necesaria al problema, pero, en cambio, aumentan el coste computacional y la dificultad de encontrar las reglas o comportamientos subyacentes a los datos. En consecuencia, la literatura que examina, propone y compara métodos de selección de variables, así como las métricas que éstos utilizan, es muy extensa. Dentro de la literatura existente cobran especial importancia y popularidad los métodos que intentan minimizar la redundancia entre variables a la vez que maximizan la relevancia con la clase a predecir. Dentro de esta categoría se enmarcan algoritmos como el *minimum Redundancy Maximum Relevance (mRMR)* y *Quadratic Programming Feature Selection (QPFS)*.

El principal objetivo de este trabajo es analizar el uso de distintas medidas de similitud entre variables aleatorias (i.e. métricas) en el algoritmo global de selección de variables QPFS. Las medidas de similitud escogidas han probado ser eficaces en otros trabajos y son la Correlación de Pearson, la Información Mutua, la Información Mutua Condicionada, la Distancia de Covarianzas y la Distancia de Correlaciones. Este trabajo realiza una estimación de la complejidad computacional que supone añadir el uso de estas medidas a QPFS y estudia el rendimiento en términos de acierto en clasificación sobre distintos conjuntos de datos de la selección de variables efectuada por QPFS cuando se utiliza Naive Bayes como clasificador. En este trabajo también se propone un nuevo algoritmo, DQPFS (Diagonal Programming Quadratic Feature Selection), para tratar de solucionar un problema de la implementación original de QPFS, que penaliza a las variables más entrópicas cuando usa la Información Mutua como medida de información, acción que puede desembocar en resultados subóptimos.

Para la realización de este TFG se ha utilizado *Python* como lenguaje de programación, así como algunas de sus librerías (*Scipy, Numpy, Pandas*). Aplicar QPFS o DQPFS requiere resolver un problema QP (Quadratic Programming), para lo que se ha empleado la librería CVXOPT. Se han realizado pruebas exhaustivas variando los parámetros inherentes a QPFS y DQPFS y a las distintas medidas de similitud.

La contribución de este TFG es un estudio de las medidas de similitud entre variables aleatorias más populares y de algunas que están ganando importancia en la literatura. Estas métricas no se habían probado con anterioridad en un algoritmo global de selección de variables. Los resultados obtenidos muestran un rendimiento positivo de las medidas de similitud propuestas, en especial de la Información Mutua Condicionada.

Palabras Clave

Selección de variables, variable redundante, variable relevante, variable irrelevante, Información Mutua, Información Mutua Condicionada, Correlación de Pearson, Distancia de Covarianzas, Distancia de Correlaciones, error de clasificación, TFG, problema QP, QPFS.

Abstract

Abstract

Feature Selection is a paradigm in the field of Machine Learning which aim is to *choose or select* the most important features in a pattern recognition problem. Nowadays, interest in feature selection methods is growing in popularity due to the appearance of high dimensional problems, in which there may be not useful and nonrelevant features. Furthermore, this kind of features increase the problem complexity and complicate the discovery of underlying rules under the data. As a consequence, the literature that examines, suggests and compares feature selection methods is extensive. Among the most popular methods are those that try to minimize redundancy between variables at the same time they maximize the relevance with the class. Examples of algorithms in this category are *minimum Redundancy Maximum Relevance (mRMR)* and *Quadratic Programming Feature Selection (QPFS)*.

The main objective of this work is to analyse the use of different similarity measures between random variables (i.e. metrics) applied in a global feature selection method, QPFS. The similarity measures used in this work have proved to be efficient in other works and include Pearson Correlation Coefficient, Mutual Information, Conditioned Mutual Information, Distance Covariance and Distance Correlation. An analysis of the computational complexity added to QPFS because of the use of this similarity measurements is also carried out in this work. Different datasets will be used in order to estimate the performance of a Naive Bayes classifier in terms of classification accuracy when the QPFS algorithm with different metrics is applied before the classifier. Due to the fact that the original implementation of QPFS penalizes entropic features, leading to suboptimal solutions, this work also suggests an algorithm, named DQPFS (Diagonal Quadratic Programming Feature Selection), to solve this problem. Exhaustive tests varying inherent parameters of QPFS and DQPFS have also been carried out.

The programming language selected to implement this work is *Python* as well as some of its packages (*Numpy, Pandas, Scipy*). Applying QPFS and DQPFS requires the optimization of a QP problem (Quadratic Programming), which is solved by *CVXOPT Python* Package.

The main contribution and novelty of this TFG is a study of the most popular similarity measures and those that are gaining popularity among literature applied to a global feature selection algorithm. The results obtained in this work show that the proposed similarity measures combined with QPFS provide competitive classification accuracies. It is especially remarkable the good performance obtained by the Conditional Mutual Information.

Key words

Feature selection, redundant festure, relevant feature, irrelevant feature, Mutua Information, Conditional Mutual Information, Pearson Correlation, Distance Covariance, Distance Correlation, classification error, TFG, QP problem, QPFS.

Agradecimientos

En primer lugar quiero agradecer a Irene Rodríguez Luján haber sido mi tutora y haber contestado con rapidez a mis dudas. Gracias por los ánimos, la positividad y prestarme tus conocimientos e ideas, nada de esto habría sido posible sin ti.

Gracias a José Ramón Berrendero, José Luis Torrecilla, Alberto Suárez y Carlos Ramos por las implementaciones y la ayuda con la Distancia de Covarianzas y la Distancia de Correlaciones.

Gracias a Carlos Santa Cruz por haberme permitido trabajar este último año en el Instituo de Ingeniería del Conocimiento y hacerme ver que, en la vida real, infinito es igual a cinco. He aprendido y disfrutado mucho durante esta experiencia.

Gracias al equipo docente de la EPS y de la Facultad de Ciencias por haberme enseñado y guiado estos últimos cinco años de mi vida. Y gracias al personal docente de mi instituto. Gracias Orlando e Israel, por haberme prestado apoyo en los momentos más difíciles de mi vida. Y gracias a Mari Ángeles, Fernando y Maribel por enseñarme a ver las Matemáticas como se merecen.

Un enorme gracias a mi familia. Gracias mamá, por la paciencia y la ayuda diaria. No sé que haría sin ti. Gracias papá, por todos los consejos y frases de ánimo, nunca los he olvidado y nunca lo haré. Y gracias Sergio, por ser siempre el lado positivo de las cosas y hacerme reír en cada momento. Gracias abuelos por vuestro apoyo y cariño incondicional.

A todos mis compañeros de aventuras estos últimos cinco años. A pesar de los momentos duros, me llevo grandes recuerdos y amigos. Gracias a mis queridas amigas, por darme siempre una vía de escape para romper con la rutina. Por último, gracias Carlos por tus ideas y contraejemplos rebuscados, por escucharme y ayudarme. Gracias, en resumen, por ser el mejor compañero.

Índice general

Índice de Figuras xi

Índice de Tablas xiii

1. Introducción 1

1.1. Motivación del proyecto . . . 1
1.2. Métodos de selección de variables. . . . 2
1.3. Objetivos y enfoque . . . 4
1.4. Metodología y plan de trabajo . . . 4

2. Selección de variables. Estado del arte 5

2.1. Medidas de similitud . . . 5
2.1.1. Coeficiente de correlación de Pearson . . . 5
2.1.2. Entropía e Información Mutua . . . 6
2.1.3. Distancia de Covarianzas . . . 8
2.1.4. Distancia de Correlaciones . . . 9
2.2. Métodos de selección de variables . . . 9
2.2.1. Métodos iterativos basados en Información Mutua . . . 9
2.2.2. Métodos globales basados en Información Mutua . . . 10

3. Modificación de QPFS 13

3.1. Matrices de diagonal dominante . . . 13
3.2. Modificación del algoritmo . . . 14
3.3. Medidas de similitud propuestas . . . 16
3.4. Análisis de complejidad computacional . . . 18

4. Experimentos y Resultados 19

4.1. Implementación . . . 19
4.2. Un ejemplo sintético . . . 20
4.3. Conjuntos de datos utilizados en los experimentos . . . 23
4.4. Comparación de las medidas utilizadas . . . 24
4.5. Complejidad temporal de las medidas de información . . . 32

5. Conclusiones y trabajo futuro **37**

5.1. Conclusiones 37

5.2. Trabajo futuro 38

Glosario de acrónimos **41**

Bibliografía **42**

A. Promedio de aciertos **47**

Índice de Figuras

1.1. Funcionamiento de un algoritmo de *filtro* de selección de variables. 2

1.2. Ejemplos de problemas de clasificación bidimensionales de dos clases. Puntos rojos y violetas reprensentan cada una de las clases. 3

1.3. Funcionamiento de un algoritmo *wrapper* de selección de variables. 4

1.4. Funcionamiento de un algoritmo de *embebido* de selección de variables. 4

2.1. Algunos ejemplos que comparan el alcance de la Correlación de Pearson y la Información Mutua. 6

2.2. Funcionamiento del algoritmo QPFS [1]. 12

4.1. Intervalos de discretización para el cálculo de la Información Mutua. 20

4.2. Conjunto de datos sintético. Puntos rojos y violetas reprensentan cada una de las clases. 21

4.3. Proyecciones del conjunto mostrado en la Figura 4.2 sobre cada posible par de variables. 22

4.4. Esquema para la evaluación del rendimiento de los métodos de selección de variables en conjuntos de datos que no tienen disponible un *dataset* de test. 23

4.5. Acierto en clasificación para el conjunto MUSK en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento. 25

4.6. Acierto en clasificación para el conjunto DIGITS en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento. 26

4.7. Acierto en clasificación para el conjunto LUNG en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento. 27

4.8. Acierto en clasificación para el conjunto WDBC en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento. 28

4.9. Acierto en clasificación para el conjunto WINE en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento. 29

4.10. Acierto en clasificación para el conjunto AUDIOLOGY en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento. 30

4.11. Acierto en clasificación para el conjunto URBAN en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento. 31

4.12. Tiempos medios de ejecución en segundos en 10 iteraciones del cálculo de la Distancia de Covarianzas y Correlaciones variando el número de muestras de un conjunto de datos sintético. 32

4.13. Tiempos medios de ejecución en segundos en 10 iteraciones del cálculo de la Información Mutua y la Información Mutua Condicionada variando el número de muestras de un ejemplo sintético. 33

4.14. Tiempos de ejecución en segundos 10 iteraciones del cálculo de la Información Mutua, la Información Mutua Condicionada y la Distancia de Covarianzas y Correlaciones variando el número de clases distintas de un conjunto de datos sintético. 34

4.15. Tiempos medios de ejecución en segundos 10 iteraciones del cálculo de Información Mutua, la Información Mutua Condicionada y la Distancia de Covarianzas y Correlaciones variando el número de muestras de un conjunto de datos sintético. 34

4.16. Tiempos medios de ejecución del algoritmo de selección de variables en 5 iteraciones. 35

A.1. Acierto en clasificación para el conjunto de datos MUSK para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud. 48

A.2. Acierto en clasificación para el conjunto de datos DIGITS para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud. 49

A.3. Acierto en clasificación para el conjunto de datos LUNG para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud. 50

A.4. Acierto en clasificación para el conjunto de datos WDBC para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud. 51

A.5. Acierto en clasificación para el conjunto de datos WINE para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud. 52

A.6. Acierto en clasificación para el conjunto de datos AUDIOLOGY para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud. . 53

A.7. Acierto en clasificación para el conjunto de datos URBAN para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud. 54

Índice de Tablas

2.1. Métodos de selección de variables iterativos basados en Información Mutua. . . . 10

2.2. Métodos de selección de variables globales basados en Información Mutua. 11

3.1. Complejidad temporal de distintas medidas de similitud entre variables y el coste de QPFS con cada una de ellas. 18

4.1. Resultados de DQPFS y QPFS con distintas medidas de similitud en el conjunto de datos de la Figura 4.2. Todos los experimentos se han hecho con $\alpha = 0.5$, $\beta = 1$ y utilizando la discretización de *Doane*. Se indica el orden de selección de las variables, el peso asignado al resolver el problema QP y su relevancia (información que comparten con la clase). 22

4.2. Descripción de los conjuntos de datos utilizados en los experimentos. 24

4.3. Acierto en clasificación en el conjunto de test para el conjunto de datos MUSK. . 25

4.4. Acierto en clasificación en el conjunto de test para el conjunto de datos DIGITS. 26

4.5. Acierto en clasificación en el conjunto de test para el conjunto de datos LUNG. . 27

4.6. Acierto en clasificación en el conjunto de test para el conjunto de datos WDBC. . 28

4.7. Acierto en clasificación en el conjunto de test para el conjunto de datos WINE. . 29

4.8. Acierto en clasificación en el conjunto de test para el conjunto de datos AUDIOLOGY. 30

4.9. Acierto en clasificación en el conjunto de test para el conjunto de datos URBAN. 31

1 Introducción

1.1. Motivación del proyecto

El Aprendizaje Automático es un campo de la Inteligencia Artificial que trata de extraer patrones y relaciones de un conjunto de datos sin necesidad de un conocimiento experto sobre ellos. Este trabajo se centrará en los problemas de aprendizaje supervisado, los llamados problemas de clasificación y regresión. Un problema de clasificación consta, básicamente, de un conjunto de datos de entrenamiento, X, etiquetados. Estas etiquetas, C, se denominan clases y, en los problemas que vamos a tratar, serán un conjunto discreto $\{c_1, ..., c_L\}$. Cada patrón de entrenamiento o ejemplo $\overline{x} \in X$ está compuesto por un conjunto de M variables, que denotaremos como X_i con $i \in \{1, ..., M\}$. Formalmente, un clasificador es una función,

$$\phi : \overline{x} \in X \rightarrow y \in \{c_1, ..., c_L\} \ ,$$

que trata de asignar a cada $\overline{x} \in X$ una etiqueta cometiendo el menor error posible. El error cometido por el clasificador se define como el número de ejemplos mal clasificados entre el total N.

Resulta bastante obvio que una elección correcta de las variables que componen nuestros datos de entrenamiento o conjunto de ejemplos es crucial para que el clasificador construido funcione correctamente. Hay variables que pueden aportar poca información al problema, o incluso impedir llegar a un clasificador óptimo. Así, siguiendo la notación empleada por A.C. Pocock en [2], las variables pueden clasificarse como relevantes, redundantes e irrelevantes (ver Figura 1.2). Las variables relevantes son las que aportan información al problema de clasificación, y las irrelevantes son aquellas que no lo hacen o causan *ruido*. Las variables redundantes son variables relevantes o irrelevantes que no aportan información *nueva* respecto a las variables anteriores. Por ejemplo, podemos estar ante un problema con una variable que representa el número de operaciones que un cliente hace en su cuenta bancaria, y otra variable que indique si el cliente ha hecho más de 10 operaciones. Obviamente, si conocemos el valor de la primera variable, también conoceremos el de la segunda, por lo que esta variable será redundante respecto a la primera (esto no quiere decir que no sea una variable relevante).

Hoy en día el problema de distinguir entre sí las variables relevantes, redundantes e irrelevantes es todo un reto, pero también una necesidad. Por ejemplo, podemos encontrar conjuntos

de datos genéticos con millares de variables, pero en los que quizá casi toda la información está contenida en un determinado gen [3]. Llevar a cabo una selección de variables hace que el problema de clasificación sea más escalable computacionalmente, y, en algunas ocasiones, pueda resolverse de forma más precisa, evitando el sobreajuste. Además, permite extraer conclusiones del conjunto de datos inicial y mejorar o simplificar la interpretabilidad del modelo final.

1.2. Métodos de selección de variables.

La literatura existente sobre la selección de variables es extensa. Puede encontrarse información útil en [4, 5]. Principalmente, se distinguen tres tipos de métodos de selección de variables en base a la interdependencia que exista entre el método de selección y el algoritmo de clasificación:

Métodos de filtro

Los métodos de filtro son algoritmos de selección de variables independientes del clasificador utilizado. Se puede ver su esquema de funcionamiento en la Figura 1.1. Estos métodos asignan una puntuación a cada variable según la información que aporta al problema, por lo que necesitan una medida de información que les permita calcular cómo de relevante es una variable (es decir, cuánta información comparte con la clase) y cómo de redundantes son dos variables (es decir, cuánta información comparten entre sí). Se hará un repaso de algunas de estas medidas en la sección 2. Estos métodos se eligen por su sencillez y mayor rapidez[5, 6]. Pueden distinguirse dos subconjuntos dentro de estos métodos de selección:

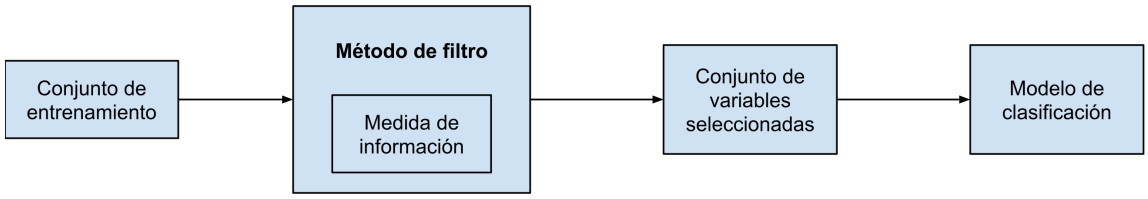
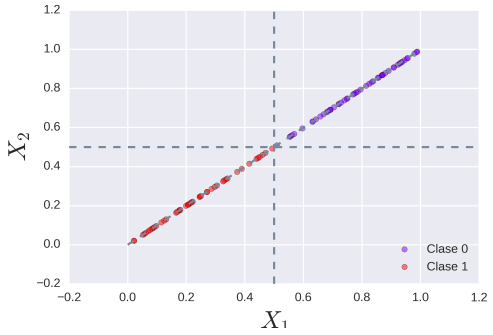


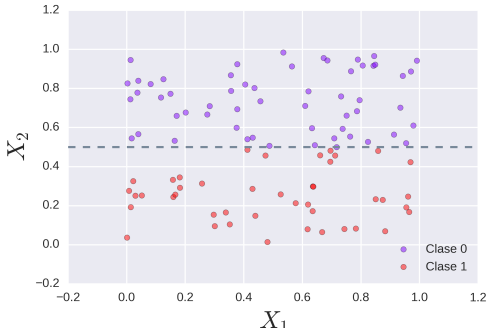
Figura 1.1: Funcionamiento de un algoritmo de *filtro* de selección de variables.

- Los algoritmos **iterativos** seleccionan en cada paso la variable más *importante* del problema dado el conjunto de variables previamente seleccionado. Este tipo de métodos puede medir sólo la información que comparte una variable con la clase, en cuyo caso nos encontraremos ante los llamados métodos de filtro univariable como MIM [7]. Si el método también tiene en cuenta la información sobre la redundancia de variables entonces nos encontraremos ante un método de filtro multivarible como mRMR [8]. Se puede encontrar una revisión más detallada de estos métodos en el Capítulo 2. Los métodos de filtro iterativos presentan, principalmente, dos inconvenientes:
  - Puede ser que un conjunto de variables sea muy relevante para nuestro problema, mientras que cada una de ellas, por separado, aporte poca información. Este tipo de variables pueden *escapar* a estos algoritmos de selección.
  - Algunas variables muy relevantes al principio pueden dejar de serlo en presencia de otras, debido a la redundancia.
- Los algoritmos de selección **global** de variables son algoritmos de un paso, es decir, tienen en cuenta todas las variables del problema a la hora de asignar una puntuación a cada variable. Ejemplos de este tipo de métodos son QPFS [1] o EQPFS [9]. Se puede encontrar una revisión más detallada de estos métodos en el Capítulo 2.

Algunos ejemplos de casuística entre la relevancia y la redundancia de variables y situaciones en las que los métodos de filtro iterativos pueden fallar se muestran en la Figura 1.2. Por ejemplo, en la Figura 1.2a es irrelevante seleccionar X_1 o X_2, mientras que en la Figura 1.2b se seleccionaría X_2. Sin embargo, en la Figura 1.2c, aunque X_1 parece ser más relevante que X_2, seleccionando ambas variables se conseguiría una clasificación perfecta. Por último, en la Figura 1.2d ambas variables parecen aportar poca información *a priori*, pero el problema es fácilmente separable si usamos ambas variables.



(a) Dos variables relevantes y redundantes entre sí.



(b) Una variable redundante (X_2) y otra irrelevante (X_1).

(c) Las dos variables son relevantes, pero X_1 parece aportar más información.

(d) Las dos variables son relevantes en igual medida.

Figura 1.2: Ejemplos de problemas de clasificación bidimensionales de dos clases. Puntos rojos y violetas reprensentan cada una de las clases.

Wrappers

Son métodos que utilizan un clasificador para puntuar variables de acuerdo a su poder de clasificación [10]. Son métodos costosos, ya que requieren entrenar un clasificador constantemente, ver Figura 1.3. Independiente del clasificador subyacente, la búsqueda de variables se puede llevar a cabo añadiendo de forma iterativa las variables más relevantes (*forward*) o, partiendo de un conjunto inicial S de variables, eliminar secuencialmente aquellas menos informativas (*backward*).

Métodos embebidos

En este tipo de métodos la selección de variables se realiza en el proceso de entrenamiento. Ver Figura 1.4. Se trata, por tanto, de clasificadores que hacen una *selección de variables* en

su fase de entrenamiento. Algunos ejemplos son los árboles de decisión [11] o las máquinas de vectores soporte [12].

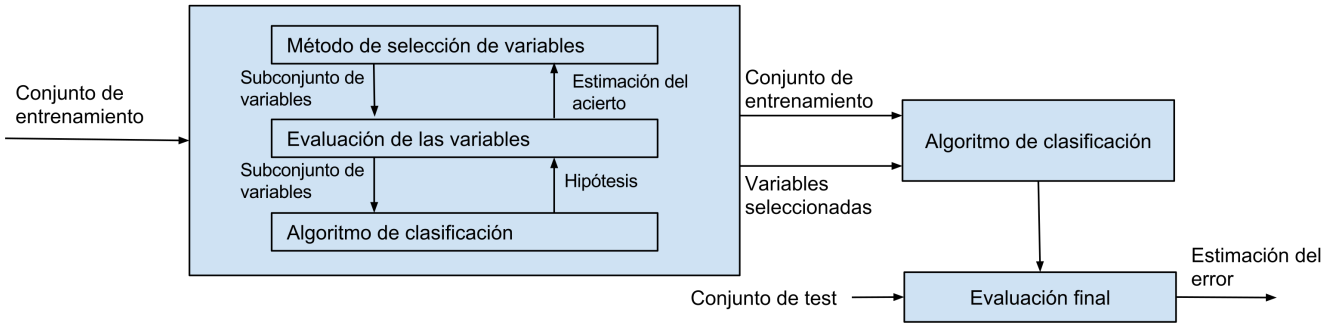


Figura 1.3: Funcionamiento de un algoritmo *wrapper* de selección de variables.

Figura 1.4: Funcionamiento de un algoritmo de *embebido* de selección de variables.

1.3. Objetivos y enfoque

El principal objetivo de este trabajo es implementar QPFS [1], un algoritmo global de selección de variables, con diferentes medidas de similitud entre variables para hacer un análisis comparativo de su rendimiento en términos de acierto en clasificación para varios conjuntos de datos. Además, se propone una solución para el problema de la entropia en la diagonal [9] de la implementación original de QPFS, que se explicará con más detalle en la Sección 2.2.2.

Para este trabajo se ha usado *Python* como lenguaje de programación, así como algunas de sus librerías. Además, se incorpora un análisis de la complejidad computacional de las medidas de similitud entre variables propuestas en este trabajo.

1.4. Metodología y plan de trabajo

Esta memoria se divide en tres capítulos, además de este capítulo de introducción. El Capítulo 2 es un repaso del estado del arte en algoritmos de selección de variables, en especial QPFS y de algunas medidas de similitud. El Capítulo 3 está dedicado a QPFS, al problema de la entropia en la diagonal y al análisis teórico del coste computacional de las diferentes medidas de similitud consideradas. El siguiente bloque, el Capítulo 4, da una breve explicación de los conjuntos de datos empleados en los experimentos y muestra los resultados obtenidos. En él se describen las pruebas realizadas y se comparan las medidas utilizadas en términos de error de clasificación y coste computacional. En el último capítulo se resumen las conclusiones derivadas de este TFG y se plantearán líneas de trabajo futuro.

2

Selección de variables. Estado del arte

En este capítulo se pretende realizar una revisión del estado del arte de las medidas de similitud entre variables aleatorias y de los métodos de filtro de selección de variables. En particular, se hace una breve revisión de las medidas de similitud que se van a emplear en los Capítulos 3 y 4. Además, este capítulo contiene una revisión de los métodos de selección de variables más utilizados, con más énfasis en QPFS [1], algoritmo en el que se centra el resto del trabajo.

2.1. Medidas de similitud

Una de las principales características de los métodos de filtro de selección de variables es que requieren una medida que cuantifique cómo de redundantes o relevantes son las variables. Son las llamadas medidas de información o similitud. Denotaremos la información que comparten dos variables aleatorias, X_1 y X_2 como $I(X_1, X_2)$. A continuación, se hace un breve repaso de algunas de las medidas existentes.

2.1.1. Coeficiente de correlación de Pearson

El coeficiente de correlación de Pearson es la primera medida a elegir como $I(X_1, X_2)$. A pesar de los inconvenientes que presenta, ya que sólo es capaz de detectar dependencias lineales entre pares de variables aleatorias, ha probado ser muy efectiva en algunos problemas de selección de atributos, tal y como se muestra en [13, 14]. El Coeficiente de Correlación de Pearson entre dos variables aleatorias, X_1 y X_2, se define como:

$$\rho(X_1, X_2) = \frac{cov(X_1, X_2)}{\sqrt{var(X_1)var(X_2)}}\,, \tag{2.1}$$

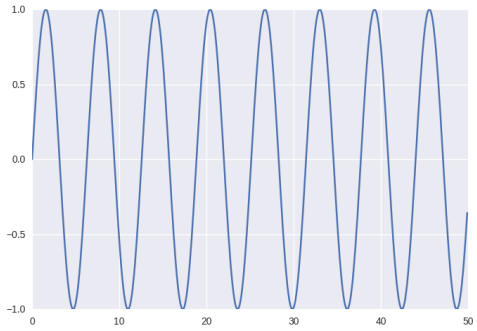
donde $cov(X_1, X_2)$ es la covarianza entre variables y $var(X_j)$ es la varianza de la variable X_j, con $j \in \{1, 2\}$.

Dada una muestra de variables las aleatorias X_1 y X_2, el coeficiente de correlación muestral se calcula como:

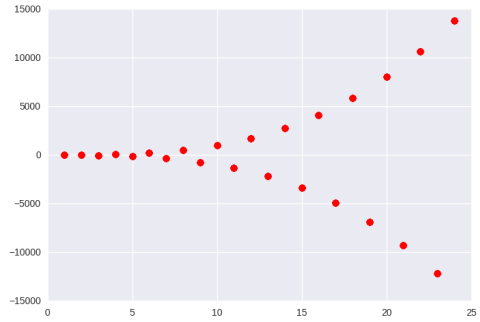
$$\hat{\rho}(X_1, X_2) = \frac{\sum_{i=1}^{N}(X_1^i - \overline{X_1})(X_2^i - \overline{X_2})}{\sqrt{\sum_{i=1}^{N}(X_1^i - \overline{X_1})\sum_{i=1}^{N}(X_2^i - \overline{X_2})}}\,, \tag{2.2}$$

donde N es el número de muestras, X_j^i es la muestra i-ésima de la variable X_j, y $\overline{X_j}$ denota la media muestral de la variable X_j, con $j \in \{1, 2\}$.

Como se ha mencionado, el coeficiente de correlación de Pearson sólo es capaz de detectar dependencias lineales entre variables. Un claro ejemplo de sus limitaciones es considerar dos variables aleatorias, X_1 y X_2 tales que $X_1 = sen(X_2)$, donde *sen* representa la función seno (ver Figura 2.1a). Claramente, el valor de X_2 depende del valor que toma X_1. Sin embargo, $\hat{\rho}(X_1, X_2) = -0.09581561$.



(a) Función seno. Aunque las variables representadas en ambos ejes son dependientes, su coeficiente de correlación es cercano a cero (0.096).



(b) Dos variables redundantes para las que la Correlación de Pearson es cercana a cero, pero no su Información Mutua.

Figura 2.1: Algunos ejemplos que comparan el alcance de la Correlación de Pearson y la Información Mutua.

2.1.2. Entropía e Información Mutua

Estas medidas nacen de la necesidad de caracterizar la independencia entre variables aleatorias. Tratan de medir la incertidumbre de una variable aleatoria y la cantidad de incertidumbre remanente cuando otra es conocida. Fueron propuestas por primera vez en [15] y han sido ampliamente utilizadas como medidas de similitud [2].

Entropía

La entropía de una variable aleatoria X mide la incertidumbre sobre un estado x de la variable X. La entropía de X se define en términos de la función de probabilidad $p(x)$ de los estados de X de la siguiente manera:

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \ . \tag{2.3}$$

Valores altos de esta medida indican que hay mucha incertidumbre sobre la variable aleatoria X. La entropía aumenta cuando aumenta el número de estados de X, que denotaremos como $|X|$, haciéndose máxima cuando todos los estados x de X son equiprobables.

La base del logaritmo define la unidad en la que se mide la entropía. Comúnmente se usa base 2 y se mide en bits, que es el estándar que se seguirá en este trabajo.

La **entropía condicionada** de la variable aleatoria X_1 dada X_2 mide, tal y como se muestra en [2], la incertidumbre esperada de X_1 cuando X_2 es conocida. Se puede definir en términos de

la probabilidad conjunta $p(x_1, x_2)$ y la condicionada $p(x_1|x_2)$ como:

$$\begin{aligned} H(X_1|X_2) &= \sum_{x_2 \in X_2} p(x_2) H(x_1|X_2 = x_2) \\ &= -\sum_{x_2 \in X_2} p(x_2) \sum_{x_1 \in X_1} p(x_1, x_2) \log p(x_1|x_2) \ . \end{aligned} \tag{2.4}$$

La entropía condicionada también puede ser definida en función de la entropía conjunta $H(X_1X_2)$ y de la entropía $H(X_2)$ como:

$$H(X_1|X_2) = H(X_1X_2) - H(X_2) \ , \tag{2.5}$$

y se puede acotar de la siguiente manera:

$$0 \leq H(X_1|X_2) \leq H(X_1) \ . \tag{2.6}$$

Información Mutua

Formalmente, la Información Mutua entre dos variables aleatorias X_1 y X_2 se define en término de las funciones de probabilidad de X_1 y X_2 como:

$$MI(X_1, X_2) = \int \int p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)} dx_1 dx_2 \tag{2.7}$$

Teorema 2.1.1. *Sean X_1, X_2 variables aleatorias. Entonces: $MI(X_1, X_2) = 0$ si y sólo si X_1 y X_2 son independientes.*

Demostración. Si X_1 y X_2 son independientes, entonces $p(x_1, x_2) = p(x_1)p(x_2)$, y, por tanto, $\log \frac{p(x_1,x_2)}{p(x_1)p(x_2)} = \log 1 = 0$. Luego $MI(X_1, X_2) = 0$. La demostración en el otro sentido es más compleja y puede consultarse en [16]. □

Para problemas discretos podemos simplificar la expresión de la Información Mutua,

$$MI(X_1, X_2) = \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} p(x_1, x_2) \log \frac{p(x_1, x_2)}{p(x_1)p(x_2)} \tag{2.8}$$

La figura 2.1b ilustra por qué puede resultar útil usar la Infomación Mutua en lugar de la Correlación de Pearson para medir la redundancia entre variables. La imagen de la figura corresponde a la función:

$$f(x) = \begin{cases} x^3 & \text{si } x \text{ es par} \\ (-x)^3 & \text{si } x \text{ es impar} \end{cases} \ ,$$

donde x puede tomar valores enteros en el intervalo $[0, 25]$ (el problema debe ser discreto). Definiendo la variable aleatoria $Y = f(X)$, se tiene que $MI(X, Y) = 3.16673936174 > 0$, mientras que $\hat{\rho}(X, Y) = 0.05960423 \simeq 0$.

La Información Mutua es una medida simétrica; es decir, $MI(X_1, X_2) = MI(X_2, X_1)$. También puede verse como una función de las entropías, tal y como se muestra en [2]. La Información Mutua será una medida de cuánto se reduce la incertidumbre de X_1 cuando se conoce X_2. Así, se obtienen las ecuaciones:

$$\begin{aligned} MI(X_1, X_2) &= H(X_1) - H(X_1|X_2) \\ &= H(X_2) - H(X_2|X_1) \\ &= H(X_1) + H(X_2) - H(X_1X_2) \ . \end{aligned} \tag{2.9}$$

La **Información Mutua Condicionada** mide la dependencia entre dos variables cuando el estado de una tercera es conocido. Puede escribirse como:

$$\begin{aligned} MI(X_1, X_2|X_3) &= \sum_{x_3 \in X_3} p(x_3) MI(X_1; X_2|X_3 = x_3) \\ &= \sum_{x_3 \in X_3} p(x_3) \sum_{x_1 \in X_1} \sum_{x_2 \in X_2} p(x_1, x_2|x_3) \log \frac{p(x_1, x_2|x_3)}{p(x_1|x_3)p(x_2|x_3)} \ . \end{aligned} \tag{2.10}$$

Aplicando la regla de la cadena, se obtiene que dadas tres variables aleatorias X_1, X_2 y X_3, se verifica que [2]:

$$MI(X_1X_2|X_3) = MI(X_1; X_3) + MI(X_2, X_3|X_1) \tag{2.11}$$

2.1.3. Distancia de Covarianzas

Esta medida fue propuesta por primera vez en [17]. La distancia de covarianzas entre dos variables aleatorias $X_1 \in \mathbb{R}^p$ y $X_2 \in \mathbb{R}^q$ es el valor no negativo definido como:

$$\nu^2(X_1, X_2) = \int_{\mathbb{R}^{p+q}} |\varphi_{X_1,X_2}(x_1, x_2) - \varphi_{X_1}(x_1)\varphi_{X_2}(x_2)|^2 w(x_1, x_2) dx_1 dx_2 \ , \tag{2.12}$$

donde $w(x_1, x_2)$ es una función de peso y φ_{X_1,X_2}, φ_{X_1} y φ_{X_2} son las funciones características de X_1X_2, X_1 y X_2 respectivamente. En [17] se elige como función de peso $w(x_1, x_2) = (c_p c_q |x_1|_p^{1+p} |x_2|_q^{1+q})^{-1}$, donde $c_d = \frac{\pi^{(1+d)/2}}{\Gamma((1+d)/2)}$ es la superficie de la semiesfera unidad en $\mathbb{R}^{d+1}$ y $|\cdot|_d$ es la norma Euclídea en $\mathbb{R}^d$.

Al igual que la Información Mutua, la Distancia de Covarianzas también caracteriza la independencia entre variables aleatorias.

Teorema 2.1.2. *Sean X_1, X_2 variables aleatorias. Entonces: $\nu^2(X_1, X_2) = 0$ si y sólo si X_1 y X_2 son independientes.*

Demostración. Si X_1 y X_2 son independientes, entonces $\varphi_{X_1,X_2} = \varphi_{X_1}\varphi_{X_2}$. Por tanto, $\nu^2(X_1, X_2) = 0$. Para la demostración completa, ver [17]. □

La Distancia de Covarianzas ya se ha usado con resultados prometedores como medida de información en algunos criterios de selección de variables avariciosos. Véase en [18] ejemplo con mRMR, algoritmo que se describe más detalladamente en la Sección 2.2.1.

Estimador de la Distancia de Covarianzas

En este trabajo utilizaremos el estimador propuesto en [19]. Dadas las muestras de las variables aleatorias X_1 y X_2, la distancia de covarianzas puede estimarse como:

$$\nu^2(X_1, X_2) = \frac{1}{N^2} \sum_{i,j=1}^{N} \hat{A}_{ij} \hat{B}_{ij} \ , \tag{2.13}$$

donde N es el número de ejemplos, $\hat{A}$ y $\hat{B}$ son las matrices de distancias doblemente centradas de X_1 y X_2 respectivamente. Es decir, dadas la matrices $(a_{ij}) = |x_1^i - x_1^j|_d$ y $(b_{ij}) = |x_2^i - x_2^j|_d$, donde $|\cdot|_d$ es la norma Euclídea en $\mathbb{R}^d$, la entrada ij de $\hat{A}$ es:

$$\hat{A}_{ij} = a_{ij} - \overline{a}_{i\cdot} - \overline{a}_{\cdot j} + \overline{a}_{\cdot\cdot} \ , \tag{2.14}$$

donde $\overline{a}_{i\cdot}$ denota la media de la fila i de (a_{ij}), $\overline{a}_{\cdot j}$ denota la media de la columna j de (a_{ij}) y $\overline{a}_{\cdot\cdot}$ es la media de todos los elementos de la matriz (a_{ij}). Estas definiciones son equivalentes para la matriz $\hat{B}$.

2.1.4. Distancia de Correlaciones

La distancia de correlaciones [17] puede considerarse la estandarización de la distancia de covarianzas. Se define como:

$$\mathcal{R}^2(X_1, X_2) = \begin{cases} \frac{\nu^2(X_1,X_2)}{\sqrt{\nu^2(X_1)\nu^2(X_2)}}, & \text{si } \nu^2(X_1)\nu^2(X_2) > 0 \\ 0, & \text{si } \nu^2(X_1)\nu^2(X_2) \leq 0 \end{cases} \tag{2.15}$$

En [19] se muestran tres propiedades interesantes de $\mathcal{R}$:

1. $0 \leq \mathcal{R} \leq 1$
2. Si $\mathcal{R}(X_1, X_2) = 1$ entonces existen $a \in \mathbb{R}^q$, $b \in \mathbb{R}$ y una matriz ortogonal R tal que $X_2 = a + bX_1R$.
3. $\mathcal{R}^2(X_1, X_2) = 0$ si y solo si X_1 y X_2 son independientes.

2.2. Métodos de selección de variables

En esta sección se pretende hacer un breve resumen de algunos de los métodos de selección de variables existentes. Nos centraremos en los métodos basados en Información Mutua, una de las medidas de información más populares entre todos los métodos de filtro [2, 5, 20].

Para aclarar la notación de esta sección supondremos que nuestro problema consta de N ejemplos o patrones, cada uno de ellos reprentado como un vector de $M + 1$ coordenadas, formadas por las M variables y la variable objetivo o clase, C. La matriz de datos que contendrá la información para el problema de clasificación será $X \in \mathbb{R}^{N\times(M+1)}$. Cuando queramos referirnos a la variable k-ésima de nuestro problema, escribiremos X_k. Si queremos referirnos al ejemplo k-ésimo, escribiremos X^k. Supondremos que la clase C puede tomar L valores distintos, siendo c_i, con $i = 1, ..., L$, sus posibles valores.

2.2.1. Métodos iterativos basados en Información Mutua

En [2, 21] puede encontrarse un marco teórico de los métodos de selección de variables iterativos o avariciosos basados en Información Mutua. A modo resumen, se incorpora la Tabla 2.1, mostrando las funciones objetivo de cada método, donde S es el conjunto de variables que ya se han seleccionado en los pasos anteriores del algoritmo.

Existen numerosos métodos de selección de variable iterativos en la literatura, aunque todos tienen algo en común: buscamos la variable X_i que, dado un conjunto de variables S previamente seleccionado aporte información útil al problema (es decir, que no sea redundante con S y esté altamente relacionada con la clase). Merece la pena destacar que, dada la función objetivo $MI(X_k, C|S)$ que trata de recoger las propiedades que se acaban de explicar, se puede probar el siguiente teorema:

Teorema 2.2.1. *Sea X_k una variable de un problema de clasificación, S un conjunto de variables previamente seleccionado y C la clase objetivo. Entonces,*

$$MI(X_k, C|S) = MI(X_k, C) - MI(X_k, S) + MI(X_k, S|C). \tag{2.16}$$

Método	Función objetivo	Ref.
MIM	$MI(C, X_i)$	[7]
MIFS	$MI(C, X_i) - \beta \sum_{X_j \in S} MI(X_i, X_j)$	[22]
mRMR	$MI(C, X_i) - \frac{1}{\|S\|} \sum_{X_j \in S} MI(X_i, X_j)$	[8]
maxMIFS	$MI(C, X_i) - max_{X_j \in S}\{MI(X_i, X_j)\}$	[23]
CIFE	$MI(C, X_i) - \sum_{X_j \in S}(MI(X_i, X_j) - MI((X_i, X_j\|C))$	[24]
JMI	$MI(C, X_i) - \frac{1}{\|S\|} \sum_{X_j \in S}(MI(X_i, X_j) - MI((X_i, X_j\|C))$	[25]
CMIM	$MI(C, X_i) - max_{X_j \in S}\{MI(X_i, X_j) - MI(X_i, X_j\|C)\}$	[26]
JMIM	$MI(C, X_i) - max_{X_j \in S}\{MI(X_i, X_j) - MI(X_i, X_j\|C) - MI(C, X_j)\}$	[27]
ICAP	$MI(C, X_i) - \sum_{X_j \in S} \text{máx}\{0, (MI(X_i, X_j) - MI((X_i, X_j\|C))\}$	[28]

Tabla 2.1: Métodos de selección de variables iterativos basados en Información Mutua. En la columna *Ref.* se proporciona la referencia al artículo principal donde se presenta cada método.

Demostración. Dada la ecuacion 2.11 se tiene:

$$MI(X_k C|S) = MI(X_k; C) + MI(X_k, S|C) \tag{2.17}$$

y, por simetría:

$$MI(X_k C|S) = MI(X_k; S) + MI(X_k, C|S) \tag{2.18}$$

Basta igualar 2.17 y 2.18 para obtener 2.16. □

Haciendo distintas asunciones sobre la independencia entre variables y con la clase, se puede llegar a muchos de los métodos mostrados en la Tabla 2.1 desarrollando la ecuación 2.16. Para más información, ver [20].

2.2.2. Métodos globales basados en Información Mutua

Esta sección presenta algunos de los métodos globales de selección de variables. Al igual que en los métodos iterativos, la literatura se centra en métodos basados en Información Mutua. En particular se presentará el algoritmo Quadratic Programming Feature Selection que trata de resolver un problema de programación cuadrática (QP), minimizando una función objetivo y devolviendo un ranking de las variables.

Problemas de programación cuadrática (QP)

Un problema de programación cuadrática o problema QP es un problema de optimización matemática que tiene la siguiente forma:

$$\begin{aligned} \min_{x \in \mathbb{R}^M} \quad & \{\frac{1}{2} x^t Q x - F^t x\} \\ \text{sujeto a} \quad & Gx \preceq h \\ & Ax = b\,, \end{aligned} \tag{2.19}$$

donde $Q \in \mathbb{R}^{M \times M}$ es una matriz cuadrada y simétrica de dimensión d, $G \in \mathbb{R}^{k \times M}$, $h \in \mathbb{R}^k$, $A \in \mathbb{R}^{k \times M}$ y $h \in \mathbb{R}^k$ son las matrices que actúan como restricciones del problema y especifican el conjunto factible $\mathcal{F}$ (i.e. el conjunto de posibles soluciones), $F \in \mathbb{R}^d$ es un vector de d componentes y $\preceq$ denota desigualdad elemento a elemento, es decir, la coordenada i-ésima de Gx debe ser menor que la coordenada i-ésima de h.

Para que el problema sea convexo y se garantice convergencia, la matrix Q debe ser semidefinida positiva (ver Definición 3.1.4). No es objeto de estudio de este trabajo profundizar en los métodos de resolución de un problema QP y se ha optado por utilizar el paquete *CVXOPT* de *Python* [29].

Funciones objetivo y métodos existentes

Se muestra un breve resumen de los métodos globales de selección de variables, con sus funciones objetivo y restricciones en la Tabla 2.2.

Método	**Función objetivo**	**Parámetros**	**Restricciones**	**Ref.**
QPFS	$\min_{x\in\mathbb{R}^M}\{\frac{1}{2}(1-\alpha)x^tQx-\alpha F^tx\}$	$Q_{ij}=I(X_i,X_j)$ $F_i=I(X_i,C)$	$x\succeq\overline{0}$ $\sum_{i=1}^M\|x_i\|=1$ $\alpha\in[0,1]$	[1]
EQPFS	$\min_{x\in\mathbb{R}^M}\{\frac{1}{2}\alpha x^t(Q-H)x-F^tx\}$	$Q_{ij}=MI(X_i,X_j)$ $H_{ij}=MI(X_i,X_j\|C)$ $F_i=MI(X_i,C)$	$x\succeq\overline{0}$ $\sum_{i=1}^M\|x_i\|=1$ $\alpha\in[0,1]$	[9]
QIP	$\max_{x\in\mathbb{R}^M}\{x^tQx\}$	$Q_{ij}=MI(X_i,X_j), j\neq i$ $Q_{ij}=MI(X_i,X_j), j=i$	$x\succeq\overline{0}$ $\|x\|_d=\sqrt{k}$	[9]

Tabla 2.2: Métodos de selección de variables globales basados en Información Mutua. En la columna *Ref.* se proporciona la referencia al artículo principal donde se presenta cada método.

Quadratic Programming Feature Selecion. QPFS.

En esta sección se tratará con más detalle QPFS, el algoritmo en el que se centra el resto del trabajo. QPFS nace como alternativa al algoritmo iterativo mRMR. A diferencia de mRMR, QPFS es un algoritmo de selección de variables global y es capaz de reducir considerablemente el coste temporal de mRMR trasladando el problema de optimización a un subespacio de menor dimensión empleando para ello el método de Nyström de diagonalización de matrices [1]. En el trabajo original se implementa QPFS y se compara su rendimiento en términos de acierto en clasificación con otros métodos de selección de variables iterativos. Las medidas que se emplearon en este trabajo se detallan a continuación:

- **Valor absoluto de la Correlación de Pearson.** Para el algoritmo QPFS, se toma Q la matriz del valor absoluto del Coeficiente de Correlación de Pearson entre las variables X_i y X_j, es decir, $(Q_{ij})=|\hat{\rho}(X_i,X_j)|$. El vector F es el vector de relevancia con la clase, y por lo tanto, $F_i=|\hat{\rho}(X_i,C)|$. Para problemas multiclase, dado que la asignación de una etiqueta arbitraria a una clase no implica una ordenación topológica de las mismas, a la hora de calcular F se utiliza el denominado Coeficiente de Correlación Ponderado de Pearson [30]. Se define la dependencia entre la variable X_i y la variable objetivo C como,

$$\hat{\rho}(X_i,C)=\sum_{j=1}^{L}p(C=c_j)|\hat{\rho}(X_i,C)|\,, \tag{2.20}$$

  donde Y es una variable que vale 1 si $C=c_j$ y 0 en otro caso.

- **Información Mutua.** En este caso, se toma Q la matriz de Información Mutua entre X_i y X_j, es decir $(Q_{ij})=MI(X_i,X_j)$. F es el vector de relevancia con la clase, y por lo tanto, $F_i=MI(X_i,C)$.

Aproximación a un subespacio de menor dimensión. En problemas de alta dimensión es muy probable que exista una alta dependencia entre las variables, haciendo difícil que la matriz Q no sea singular. Además, aunque utilizando como medida de información la Correlación de Pearson, Q resulte una matriz definida positiva, no ocurre lo mismo con su valor absoluto, la Información Mutua o, en principio, con cualquier otra medida de información simétrica que desee utilizarse para rellenar la matriz Q, violando el requisito de que Q sea semidefinida positiva en un problema QP. En [1] estos problemas se resuelven trasladando y resolviendo el problema en un subespacio convexo de menor dimensión. El esquema del funcionamiento del algoritmo original se muestra en la figura 2.2.

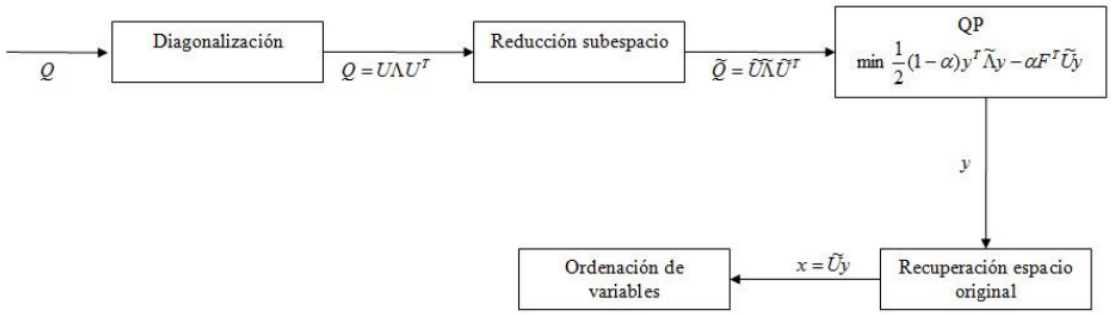


Figura 2.2: Funcionamiento del algoritmo QPFS [1], donde Λ y U son las matrices de autovalores y autovectores de Q respectivamente, es decir, $Q = U\Lambda U^t$. $\widetilde{\Lambda}$ es la matriz formada por las k primeras filas y columnas de Λ (autovalores positivos que garantizan una aproximación convexa al problema), $\widetilde{U}$ es una matriz $M \times k$ con los primeros k autovectores de Q y $y = \widetilde{U}^t x$ es un vector en $\mathbb{R}^k$.

En el artículo original se da una cota para el error cometido en la solución del problema QP al realizar esta reducción del espacio.

El problema de la entropía en la diagonal. Como se muestra en [9], QPFS penaliza a las variables por su *autoredundancia* en la matriz Q. Es fácil probar que $Q_{ii} = MI(X_i, X_i) = H(X_i)$, por lo que QPFS penaliza a las variables más entrópicas. Sin embargo, que una variable tenga la entropía alta no quiere decir que no sea relevante en el problema de clasificación. Desgracidamente, hacer $Q_{ii} = 0$ no es una opción, ya que se rompe el requisito de que Q sea una matriz definida positiva. Este problema ya fue detectado en [9] y se da una posible solución en el Capítulo 3.

3

Modificación de QPFS

En este capítulo se aborda en detalle la modificación que se ha hecho de QPFS y se presentan las las nuevas medidas de similitud que se usarán en los experimentos del Capítulo 4.

En primer lugar, con ánimo de dar una solución al problema de la entropía en la diagonal citado en la Sección 2.2.2, se trata el tema de las matrices de diagonal dominante, dando una definición, ejemplos y destacando algunas interesantes propiedades de estas matrices. A continuación, se explica cómo se han introducido en el algoritmo de QPFS estos resultados y el posible impacto en la clasificación final de las variables.

La última sección contiene la lista detallada de las medidas de similitud consideradas para su inclusión en QPFS, así como un estudio teórico de la complejidad temporal de las mismas.

3.1. Matrices de diagonal dominante

A continuación se dan una serie de definiciones y resultados matemáticos que resultarán útiles para la sección 3.2 del trabajo.

Definición 3.1.1. *Una matriz cuadrada $A = (a_{ij})$ se dice* ***matriz de diagonal dominante por filas*** *si, para cada fila de A, la magnitud de la entrada de la diagonal supera a la suma de las magnitudes de los elementos del resto de la fila. Más formalmente:*

$$|a_{ii}| \geq \sum_{i \neq j} |a_{ij}| \tag{3.1}$$

Por ejemplo, la matriz $\begin{bmatrix} 6 & -2 & 3 \\ 1 & 5 & 3 \\ -2 & 0 & 2 \end{bmatrix}$ es una matriz de diagonal dominante por filas.

Definición 3.1.2. *Se dice que una matriz $A = (a_{ij}) \in \mathbb{C}^{n \times n}$ es* ***Hermitiana*** *(o* ***Hermítica****) si es igual a su propia traspuesta conjugada. Es decir, $A = \overline{A^t}$.*

Observación 3.1.3. *Cualquier matriz real simétrica es hermítica, ya que si A es una matriz real, entonces $A = \overline{A}$.*

Definición 3.1.4. *Se dice que una matriz cuadradra $A \in \mathbb{R}^{n\times n}$ es **definida positiva** si $\forall x \in \mathbb{R}^n$ se verifica que $x^tAx > 0$ (esto es equivalente a que si λ es un autovalor de A, entonces $\lambda > 0$).*

*Se dice que A es **semidefinida positiva** si $\forall x \in \mathbb{R}^n$ se verifica que $x^tAx \geq 0$ (esto es equivalente a que si λ es un autovalor de A, entonces $\lambda \geq 0$).*

Definición 3.1.5. *Se dice que una matriz es no singular si su determinante es distinto de cero (i.e. es invertible).*

Teorema 3.1.6. *Sea $A \in \mathbb{R}^{n\times n}$ una matriz hermítica de diagonal dominante por filas. Entonces, A es semidefinida positiva.*

Demostración. La demostración del Teorema 3.1.6 puede verse en [31]. □

3.2. Modificación del algoritmo

Según lo mostrado en la Sección 2.2.2, QPFS genera un *ranking* de las variables según el peso que es asignado a cada una de ellas al resolver el problema:

$$\begin{aligned} \min_{x\in\mathbb{R}^M} \quad & \{\frac{1}{2}(1-\alpha)x^tQx - \alpha F^tx\} \\ \text{sujeto a} \quad & x \succeq \overline{0} \\ & \sum_{i=1}^{M} |x_i| = 1 \ , \end{aligned} \tag{3.2}$$

donde $Q_{ij} = I(X_i, X_j)$, $F_i = I(X_i, C)$, $\alpha \in [0, 1]$, e I una medida de información simétrica.

Solución al problema de la entropía en la diagonal. Como se discutió en la Sección 2.2.2, el algoritmo original de QPFS presenta algunos problemas cuando se usa la Información Mutua como medida de Información entre variables aleatorias, penalizando a las más entrópicas [9]. La solución que se propone es la siguiente modificación de la matriz Q:

$$Q_{ij} = \begin{cases} MI(X_i, X_j) \text{ si } i \neq j \\ \gamma \text{ si } i = j \end{cases} \ , \tag{3.3}$$

con $\gamma \in \mathbb{R}$ lo bastante grande para que Q sea una matriz diagonal dominante por filas y, por lo tanto, semidefinida positiva. Es importante que γ permanezca constante en toda la diagonal, de forma que todas las variables sufran la misma penalización por la *autoredundancia*. De esta manera se consigue eliminar la entropía de la diagonal sin violar la restricción de que Q debe ser semidefinida positiva, garantizando que seguimos ante un problema convexo y se tenga convergencia.

Debe destacarse que, aunque todas las variables sean penalizadas equitativamente, la introducción de este parámetro no nos lleva necesariamente a la misma ordenación de variables, ya que:

$$\begin{aligned} \min_{x\in\mathbb{R}^M} \{\frac{1}{2}(1-\alpha)x^t(Q+\gamma\mathcal{I})x - \alpha F^tx\} &= \min_{x\in\mathbb{R}^M} \{\frac{1}{2}(1-\alpha)x^tQx + x^t\gamma\mathcal{I}x - \alpha F^tx\} \\ &= \min_{x\in\mathbb{R}^M} \{\frac{1}{2}(1-\alpha)x^tQx + \gamma x^t\mathcal{I}x - \alpha F^tx\} \\ &= \min_{x\in\mathbb{R}^M} \{\frac{1}{2}(1-\alpha)x^tQx + \gamma||x||_2^2 - \alpha F^tx\} \ , \end{aligned} \tag{3.4}$$

donde $\mathcal{I}$ denota la matriz identidad de orden M y $||x||_2^2$ es la Norma Euclídea del vector x.

Como se observa en la ecuación 3.4, al introducir el parámetro γ, se ha introducido un término de penalización a la norma del vector x, $||x||_2^2$, que puede llevar a una ordenación diferente de las variables.

Elección del parámetro γ. Una vez que se han calculado todas las entradas de la matriz Q a excepción de los elementos de la diagonal, el cálculo de γ que se propone en este trabajo es el siguiente:

$$\gamma = \max_i \left\{ \sum_{\substack{j=1 \\ j \neq i}}^{M} |Q_{ij}| \right\}, i \in \{1, ..., M\} \ . \tag{3.5}$$

Cálculo del parámetro α. En la publicación original [1], los autores proponen una heurística para el cálculo del parámetro α. La idea es, dado que algunas medidas de información no están acotadas, equiparar el peso que tienen Q y F en el problema de optimización QP. Este α empírico, $\hat{\alpha}$, se calcula como:

$$\hat{\alpha} = \frac{\overline{q}}{\overline{q} + \overline{f}} \ , \tag{3.6}$$

donde $\overline{q}$ y $\overline{f}$ son la media de los elementos de Q y F respectivamente. Es decir,

$$\overline{q} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} Q_{ij} \ , \tag{3.7}$$

$$\overline{f} = \frac{1}{M} \sum_{i=1}^{M} F_i \ . \tag{3.8}$$

En la nueva versión del algoritmo se ha decidido mantener este α empírico, con la excepción de que, para calcular $\overline{q}$, no se tendrán en cuenta los elementos de la diagonal. Es decir,

$$\hat{\hat{\alpha}} = \frac{\hat{\overline{q}}}{\hat{\overline{q}} + \overline{f}} \ , \tag{3.9}$$

donde $\overline{f}$ se mantiene como en la Ecuación 3.8 y

$$\hat{\overline{q}} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} \widetilde{Q}_{ij} \ , \tag{3.10}$$

donde $\widetilde{Q}_{ij} = Q_{ij}$ si $j \neq i$ y 0 en otro caso.

Esto es porque, en algunas ocasiones, γ puede tomar valores muy elevados, lo que supone $\overline{q} \gg \overline{f}$ y, en consecuencia, $\alpha \simeq 1$. Esto supone que el algoritmo de mucha importancia a la relevancia (es decir, al vector F) y apenas tenga en cuenta la redundancia entre variables (es decir, ignore Q).

Algoritmo propuesto El pseudocódigo que incluye las modificaciones mencionadas de la nueva implementación de QPFS, DQPFS (Diagonal Quadratic Programming Feature Selection), se muestra en el Algoritmo 1. No está en el ámbito de este TFG realizar un estudio de los distintos métodos de discretización de variables continuas dado que elegir el método de discretización adecuado no es una tarea sencilla. El algoritmo de discretización elegido para los experimentos de este TFG se detalla en el Capítulo 4.

Algoritmo 1 Diagonal Quadratic Programming Feature Selection

Entrada: X, matriz de datos
y, etiquetas
α, parámetro
discr, bandera discretización
ddm, bandera matriz diagonal dominante
G, h, A, b, matrices con las restricciones de QPFS
Salida: weights, array con los pesos asignados a cada variable.

```
si discr es cierto entonces
    discretizar(X)
fin si
Q , F = calcularInformacion(X, y)
si ddm es cierto entonces
    convertirEnDiagonalDominante(Q)
fin si
si α es none entonces
    α = calcularAlphaEmpirico(Q, F)
fin si
devolver resolverQP (αQ, (1 − α)F, G, h, A, b)
```

3.3. Medidas de similitud propuestas

DQPFS admite cualquier medida de similitud entre variables siempre y cuando la matriz Q sea simétrica. En este trabajo, se han probado como medidas de similitud el valor absoluto de la correlación de Pearson (sección 2.2.2) y la Informacion Mutua (sección 2.1.2), que son las medidas publicadas en el algoritmo original [1]. Además, se han añadido la Información Mutua Condicionada (sección 3.3), la Distancia de Covarianzas (sección 2.1.3) y la Distancia de Correlaciones (sección 2.1.4), dado que han demostrado buen rendimiento en otros algoritmos de selección de variables [2, 18, 21].

Información Mutua Condicionada

A la hora de asignar un peso a la variable X_k en un problema de clasificación puede resultar útil tener en cuenta el término $MI(X_k, S|C)$. Este término ya se ha introducido en algunos métodos de selección de variables iterativos [2, 24, 26, 27, 25, 28], y se puede establecer una analogía para un algoritmo global de selección de variables. Para incluir este término se propone la siguiente modificación de QPFS:

$$\begin{aligned} \min_{x \in \mathbb{R}^M} \quad & \{\frac{1}{2}(1-\alpha)x^t\big(Q-\beta H\big)x - \alpha F^t x\} \\ \text{sujeto a} \quad & x \succeq \overline{0} \\ & \sum_{i=1}^{M} |x_i| = 1 \ , \end{aligned} \tag{3.11}$$

donde α y β son parámetros en el intervalo $[0, 1]$, F es el vector de relavancia (similitud de cada variable con con la clase), Q es la matriz de redundancias (similitud entre cada par de variables) y H es la matriz de redundancia condicionada a la clase (similitud entre cada par de variables condicionado a que conocemos el valor de la clase). Es decir,

$$F_i = MI(X_i, C) \ , \tag{3.12}$$

$$Q_{ij} = MI(X_i, X_j) \ , \tag{3.13}$$

$$H_{ij} = MI(X_i, X_j|C) \ . \tag{3.14}$$

Tomando $\beta = 0$ recuperamos el algoritmo original QPFS con Información Mutua. Al igual que en la formulación original de QPFS, tenemos dos opciones para rellenar la diagonal de las matrices Q y H:

- Si tomamos $Q_{ii} = H_{ii} = 0$, entonces la matriz es indefinida y no se garantiza encontrar un mínimo en la Ecuación 3.11.
- Si tomamos $Q_{ii} = H(X_i)$ y $H_{ii} = MI(X_i, X_i|C) = H(X_i|C)$, a excepción de β, aparece en la diagonal el término $H(X_i) - H(X_i|C) = MI(X_i, C)$. Es decir, se penaliza a las variables por su relevancia con la clase, lo cual claramente no es deseable.

La solución es, al igual que en el apartado anterior, convertir a la matriz $\big(Q - \beta H\big)$ en una matriz de diagonal dominante por filas. Por lo tanto,

$$\big(Q - \beta H\big)_{ij} = \begin{cases} MI(X_i, X_j) - \beta MI(X_i, X_j|C) \text{ si } i \neq j \\ \gamma \text{ si } i = j \end{cases} \ , \tag{3.15}$$

donde γ es lo bastante grande para que $\big(Q - \beta H\big)$ sea una matriz semidefinida positiva.

Distancia de Covarianzas

La medida mostrada en 2.1.3 tiene la ventaja de que también, al igual que la Información Mutua, es capaz de detectar dependencias no lineales entre variables. Para el algoritmo QPFS tomaremos la matriz Q como,

$$Q_{ij} = \nu^2(X_i, X_j) \ , \tag{3.16}$$

y el vector de relevancias, F, como,

$$F_i = \nu^2(X_i, C) \ . \tag{3.17}$$

Distancia de Correlaciones

La versión de QPFS con esta medida similitud tomaría la matriz Q como,

$$Q_{ij} = \mathcal{R}^2(X_i, X_j) \ , \tag{3.18}$$

y el vector de relevancias, F, como,

$$F_i = \mathcal{R}^2(X_i, C) \ . \tag{3.19}$$

No es objeto de este TFG estudiar si la Distancia de Covarianzas y la Distancia de Correlaciones generan una matriz Q definida positiva. Por lo tanto, se ha decidido convertir Q en una matriz diagonal dominante por filas.

3.4. Análisis de complejidad computacional

En esta sección se pretende hacer un análisis de la complejidad computacional de las medidas de información descritas en el apartado anterior. A modo resumen se incorpora la Tabla 3.1. No es objeto de este TFG hacer un análisis de la complejidad de QPFS contra otros métodos de selección de variables, sino dar una comparativa de QPFS o DQPFS cuando se aplican distintas medidas de similitud entre variables. En [1] se estima que el coste de QPFS, suponiendo una medida de información con coste lineal en el número de patrones como la Correlación de Pearson o la Información Mutua, es $\mathcal{O}(NM^2)$ si $N \gg M$. En otro caso, el rendimiento es $\mathcal{O}(M^3)$, independientemente del número de ejemplos. Para las medidas de información utilizadas, tenemos lo siguiente:

- **Información Mutua.** El coste de calcular la Información Mutua de dos variables X_1 y X_2 con N ejemplos depende de la cantidad de valores distintos que puedan tomar ambas variables aleatorias. Si X_1 y X_2 pueden tomar n_1 y n_2 valores distintos respectivamente, el coste de calcular la $MI(X_1, X_2)$ será $\mathcal{O}(n_1 n_2 N)$. Fijando un método de discretización, n_1 y n_2 serán fijos, y tendremos un coste $\mathcal{O}(N)$. Es decir, lineal en el número de ejemplos.
- **Información Mutua Condicionada.** La complejidad de calcular la Información Mutua Condicionada es similar a la de calcular la Información Mutua, aunque tenemos que iterar en el número de clases distintas L que puede tomar la variable objetivo C. Por lo tanto, el coste de calcular $MI(X_1, X_2|C)$ será $\mathcal{O}(n_1 n_2 NL)$. Igual que en el caso anterior, fijando n_1 y n_2, tendremos un coste $\mathcal{O}(LN)$.
- **Distancia de Covarianzas.** Según lo visto en la sección 2.1.3 sobre el estimador de la Distancia de Covarianzas, hay que calcular el doble centrado de las matrices de distancias $\hat{A}$ y $\hat{B}$. El coste de esta operación, si X_1 y X_2 tienen dimensión d, es $\mathcal{O}(dN^2)$. En nuestro caso, $d = 1$, y, por lo tanto, el coste es $\mathcal{O}(N^2)$. Una vez se han calculado $\hat{A}$ y $\hat{B}$, hay que hacer la media de $\hat{A} * \hat{B}$, donde $*$ denota la multiplicación elemento a elemento. El coste de esta operación es $\mathcal{O}(N^2)$. Por lo tanto, el coste de calcular la Distancia de Covarianzas es cuadrático en el número de ejemplos, $\mathcal{O}(N^2)$.
- **Distancia de Correlaciones.** En el cálculo de la Distancia de Correlaciones de dos variables, X_1 y X_2, intervienen la Distancia de Covarianzas de X_1, X_2 y X_1X_2. Por lo tanto, el coste es $\mathcal{O}(N^2)$.

Medida	**Coste Medida**	**QPFS + Medida ($N \gg M$)**	**QPFS + Medida ($N << M$)**
MI	$\mathcal{O}(N)$	$\mathcal{O}(NM^2)$	$\mathcal{O}(M^3)$
CMI	$\mathcal{O}(LN)$	$\mathcal{O}(LNM^2)$	$\mathcal{O}(M^3)$
DCOV	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2M^2)$	$\mathcal{O}(M^3)$
DCOR	$\mathcal{O}(N^2)$	$\mathcal{O}(N^2M^2)$	$\mathcal{O}(M^3)$

Tabla 3.1: Complejidad temporal de distintas medidas de similitud entre variables y el coste de QPFS con cada una de ellas. N representa el número de ejemplos, M el número de variables y L el número de valores distintos que puede tomar la variable objetivo C (número de clases).

4

Experimentos y Resultados

En este capítulo se compara empíricamente el algoritmo QPFS (Quadratic Programming Feature Selection) con la modificación propuesta, DQPFS (Diagonal Quadratic Programming Feature Selection), y las medidas de similitud explicadas en el Capítulo 3. En primer lugar se abordan detalles de la implementación de los algoritmos, el método de discretización utilizado para los atributos continuos en las medidas que así lo requieren (Información Mutua e Información Mutua Condicionada), y el clasificador que se usará para evaluar los distintos conjuntos de variables generados por los métodos de selección. En segundo lugar se hace un análisis del coste computacional real de las medidas de información utilizadas, corroborando los resultados teóricos de la Sección 3.4. En tercer lugar, se presenta y analiza un ejemplo sintético para dar luz a la aportación que hacen las nuevas medidas de similitud al algoritmo original. En la siguiente sección se da una explicación detallada de los conjuntos de datos del mundo real que se usan en los experimentos y el método de evaluación utilizado. Finalmente, en la última sección, se exponen y analizan los resultados obtenidos en términos de error de clasificación y coste computacional.

4.1. Implementación

La totalidad del código se ha implementado usando *Python* como lenguaje de programación. Las librerías utilizadas han sido las siguientes:

- NUMPY [32] para la carga de los conjuntos de datos y distintas operaciones con matrices (media, desviación típica, coeficiente de correlación), cálculo de autovalores y autovectores, diagonalización, producto escalar y trasposición de matrices.
- SCIKIT-LEARN [33] para la implementación de la Información Mutua y distintos clasificadores. Como QPFS es un método de filtro (independiente del clasificador), por simplicidad, se ha usado Naive-Bayes en los experimentos. Además, es un clasificador especialmente interesante para probar métodos de selección de variables que intentan minimizar la redundancia, ya que supone independencia entre variables.
- CVXOPT [29] para la resolución de problemas QP.

- SCIPY [34] para calcular las matrices de distancias de la Distancia de Covarianzas y de Correlaciones. También se han usado de este paquete PANDAS y MATPLOTLIB para el almacenamiento y representación gráfica de resultados y ejemplos respectivamente.

Discretización de los datos Como vimos en la Sección 2.1.2, el cálculo de la Información Mutua requiere estimar funciones de densidad. En el caso de variables continuas se ha decidido usar la misma discretización que en el artículo original de QPFS [1] y discretizar las mismas en tres segmentos, tal y como se muestra en la Figura 4.1. Dadas la desviación típica σ y la media μ de una variable aleatoria continua X_i, suponiendo que los datos siguen una distribución normal $\mathcal{N}(\mu, \sigma)$, consideramos tres segmentos, a saber, $(-\infty, \mu - \sigma)$, $[\mu - \sigma, \mu + \sigma)$ y $[\mu + \sigma, \infty)$.

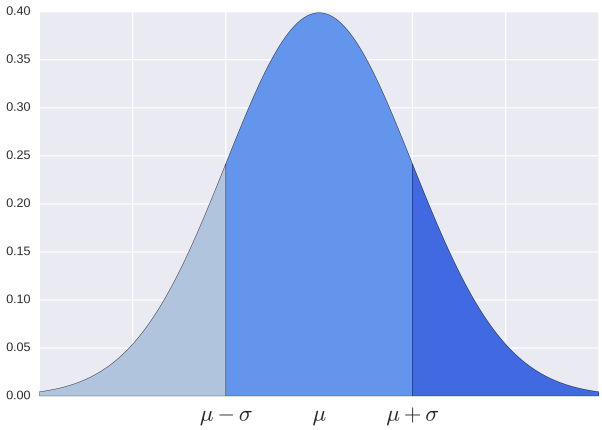


Figura 4.1: Intervalos de discretización para el cálculo de la Información Mutua.

En el ejemplo de la Sección 4.2 se ha preferido discretizar el conjunto de datos con la fórmula para dibujar histogramas de *Doane* [35]. Esto es porque la discretización propuesta en la Figura 4.1 es no es suficientemente fina y, aunque ofrece resultados similares, no permite llegar al nivel de detalle que se desea en este ejemplo. La discretización de Doane sirve tanto para una distribución normal como para otras y establece el número de intervalos en los que se van a discretizar los datos, k, como:

$$k = 1 + \log_2 N + \log_2 \left(1 + \frac{|g_1|}{\sigma_{g_1}} \right) , \tag{4.1}$$

donde g_1 es el tercer momento o asimetría de los datos y

$$\sigma_{g_1} = \sqrt{\frac{6(N-2)}{(N+1)(N+3)}} \ . \tag{4.2}$$

Después, se hacen k intervalos de la misma longitud.

4.2. Un ejemplo sintético

Este capítulo pretende dar un ejemplo del alcance de algunas de las medidas de similitud propuestas en la Sección 3.3, en especial de la Información Mutua Condicionada. Se ha generado un conjunto de datos de $M = 3$ variables y $N = 500$ ejemplos tal y como se muestra en la Figura

4.2. En este conjunto, las variables X_1 y X_2 permiten una clasificación perfecta (ver Figura 4.3a). Sin embargo, la variable X_3 puede parecer relevante, ya que los ejemplos morado están ligeramente por debajo de los rojos si sólo se tiene en cuenta esta variable (ver Figuras 4.3b y 4.3c). Se incluyen las proyecciones de este conjunto sobre el espacio bidimensional formado por cada posible par de variables en la Figura 4.3.

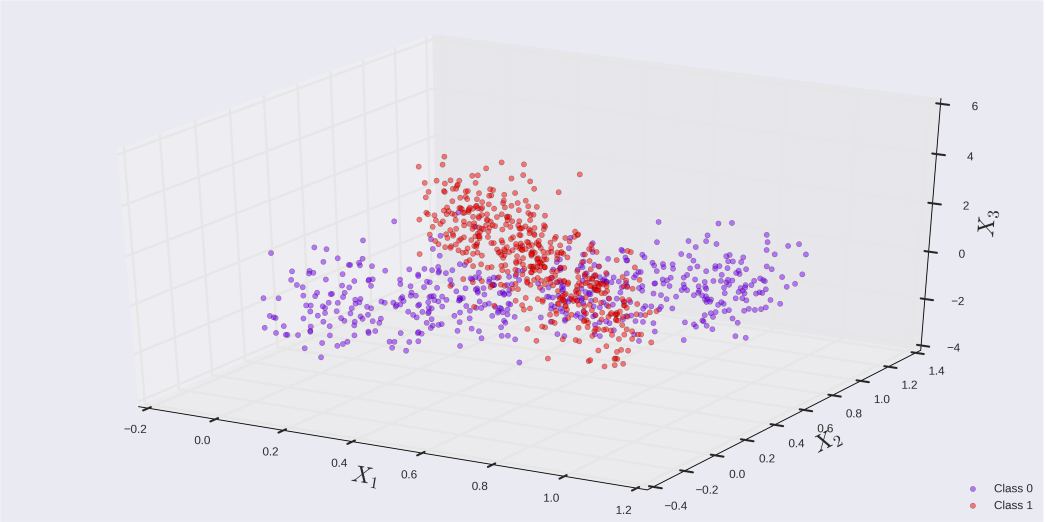
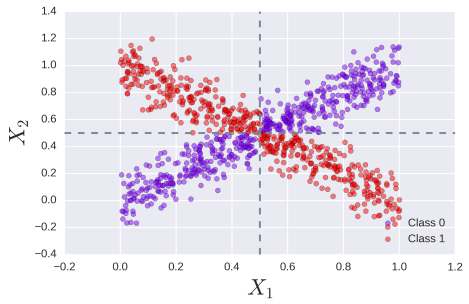
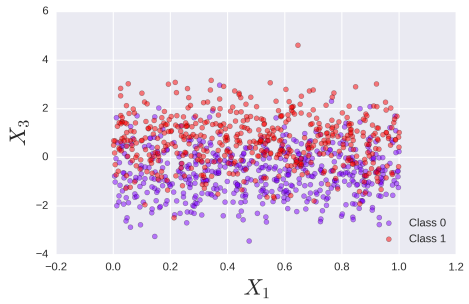


Figura 4.2: Conjunto de datos sintético. Puntos rojos y violetas reprensentan cada una de las clases.

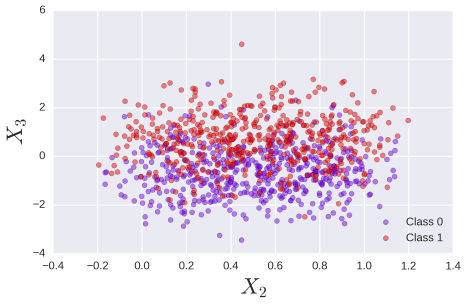
Los resultados obtenidos en los experimentos se muestran en la Tabla 4.1. La mayoría de las medidas asignan mucho peso a la variable X_3, ya que es la variable que más información comparte con la clase, independientemente de la medida utilizada. Sin embargo, X_1 y X_2 consiguen una clasificación perfecta cuando actúan en conjunto y sólo el método DQPFS con la Información Mutua Condicionada asigna más peso a estas variables que a X_3. Esto se debe a que individualmente las variables son irrelevantes, ya que $I(X_1, C) \approx I(X_2, C) \approx 0$. Además, la redundancia entre X_1 y X_2 es pequeña, ya que para valores pequeños de X_1, X_2 alcanza indistintamente cualquier valor entre su máximo y su mínimo y viceversa. Sin embargo, $MI(X_1, X_2|C) \approx 1$, lo que hace que los pesos asignados a cada variable cambien. Como se observa en la Tabla 4.1, el orden de variables proporcionado por DQPFS usando la Información Mutua Condicionada es capaz de detectar la importancia del par (X_1, X_2) para clasificar los datos de las Figuras 4.2 y 4.3.



(a) Proyección sobre el plano de X_1 y X_2.



(b) Proyección sobre el plano de X_1 y X_3.



(c) Proyección sobre el plano de X_2 y X_3.

Figura 4.3: Proyecciones del conjunto mostrado en la Figura 4.2 sobre cada posible par de variables.

Medida de similitud	**Clasificación**	**Peso asignado**	**Relevancia**
QPFS + Correlación de Pearson	X_3	0.693933	0.5511284
	X_1	0.178131	0.038631
	X_2	0.127936	0.004958
DQPFS + Correlación de Pearson	X_3	1.000000e+00	0.551128
	X_1	2.413232e-09	0.038631
	X_2	1.949499e-09	0.004958
QPFS + Información Mutua	X_3	0.459178	0.177301
	X_2	0.281704	0.004763
	X_1	0.259118	0.004841
DQPFS + Información Mutua	X_3	0.652985	0.177301
	X_2	0.222240	0.004763
	X_1	0.124775	0.004841
DQPFS + Información Mutua Condicionada	X_1	0.357073	0.004841
	X_2	0.356358	0.004763
	X_3	0.286569	0.177301
QPFS + Distancia de Covarianzas	X_3	1.000000e+00	0.119350
	X_2	9.528938e-09	0.000039
	X_1	9.488954e-09	0.000158
QPFS + Distancia de Correlaciones	X_3	1.000000e+00	0.301713
	X_1	1.369141e-08	0.001525
	X_2	1.361920e-08	0.000365

Tabla 4.1: Resultados de DQPFS y QPFS con distintas medidas de similitud en el conjunto de datos de la Figura 4.2. Todos los experimentos se han hecho con $\alpha = 0.5$, $\beta = 1$ y utilizando la discretización de *Doane*. Se indica el orden de selección de las variables, el peso asignado al resolver el problema QP y su relevancia (información que comparten con la clase).

4.3. Conjuntos de datos utilizados en los experimentos

Los conjuntos de datos utilizados en este trabajo se muestran en la Tabla 4.2 junto con el promedio del porcentaje de error sin realizar selección de variables utilizando como clasificador Naive Bayes [14] y validación cruzada con 5 *folds* siguiendo el esquema de la Figura 4.4. En la Sección 4.4 y en el Anexo A se repesentan las curvas de error en función del número de variables con distintos valores de los parámetros α y β. Las pruebas se han hecho con $\alpha \in \{0.25, 0.5, 0.75\}$ y el α empírico, $\hat{\alpha}$, descrito en la Sección 3.2, y con $\beta \in \{0.25, 0.5, 0.75, 1\}$. De todos los conjuntos utilizados que no disponían en la fuente original de descarga de un conjunto de *test* para realizar las pruebas se ha extraído uno aleatoriamente que suponía un 20 % del total de ejemplos N.

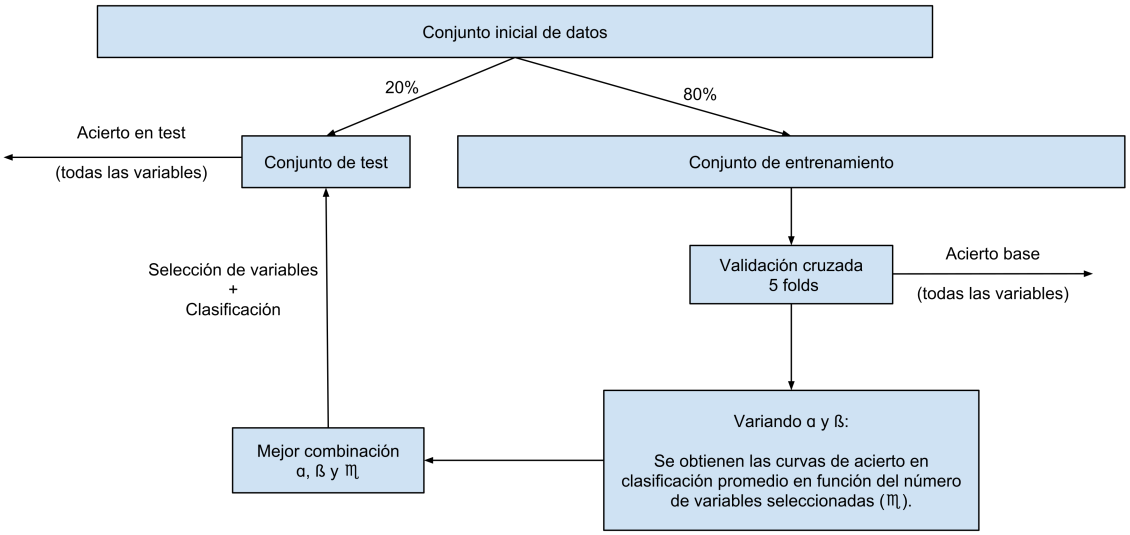


Figura 4.4: Esquema para la evaluación del rendimiento de los métodos de selección de variables en conjuntos de datos que no tienen disponible un *dataset* de test.

A continuación se proporciona información descriptiva de los conjuntos de datos:

- **LUNG**: conjunto de datos con muestras de tejido pulmonar clasificados en siete clases. Este *dataset* está previamente normalizado y discretizado en tres intervalos. Puede encontrarse en `http://home.penglab.com/proj/mRMR/#online`.
- **DIGITS**: conjunto de datos creado para la asignatura de *Fundamentos de aprendizaje automático* en la Universidad Autónoma de Madrid, 2017. Contiene información sobre los píxeles de imágenes con dígitos manuscritos del 0 al 9 y 10 clases.
- **WINE**: conjunto de datos muy sencillo con resultados del análisis químico de vinos cultivados en la misma región de Italia, de tres cultivos diferentes. Puede encontrarse en el repositorio *UCI Machine Learning*, en `https://archive.ics.uci.edu/ml/datasets/wine`.
- **WDBC**: conjunto de datos con información descriptiva de los núcleos de células en presencia y ausencia de cáncer de mama. Puede encontrarse en el repositorio *UCI Machine Learning*, en `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`.
- **MUSK**: conjunto de datos con información 92 móleculas de las cuales 47 son juzgadas por un humano experto como *musk*. Puede encontrarse en el repositorio *UCI Machine Learning*, en `https://archive.ics.uci.edu/ml/datasets/Musk+%28Version+1%29`.

- **URBAN**: conjunto de datos para clasificar una imagen aérea de alta definición en 9 tipos de territorio. Tiene un número bajo de ejemplos de entrenamiento para cada clase (14-30) y una gran cantidad de variables (148), por lo que es un problema especialmente interesante para aplicar selección de variables. Puede encontrarse en el repositorio *UCI Machine Learning*, en `https://archive.ics.uci.edu/ml/datasets/Urban+Land+Cover` y hay disponible un conjunto de entrenamiento y otro de test.
- **AUDIOLOGY**: conjunto de datos multiclase previamente estandarizados. Está formado por distintas características de pacientes y su correspondiente diagnóstico auditivo. Pueden encontrarse en el repositorio *UCI Machine Learning* un conjunto de entrenamiento y otro de test, en `https://http://archive.ics.uci.edu/ml/datasets/audiology+(standardized)`.

Nombre	Tipo de dato	N	M	L	Error	Disponible test	Referencias
LUNG	Discreto	73	325	7	0.8257	NO	[36]
DIGITS	Continuo	940	209	10	0.4892	NO	
WINE	Continuo	178	13	3	0.9857	NO	[37, 38]
WDBC	Continuo	569	30	2	0.9296	NO	[39, 40]
MUSK	Entero	476	166	2	0.7052	NO	[41]
URBAN	Continuo	169	148	9	0.7923	SÍ	[42]
AUDIOLOGY	Discreto	226	69	24	0.6153	SÍ	[43]

Tabla 4.2: Descripción de los conjuntos de datos utilizados en los experimentos. N representa el número de ejemplos, M el número de variables y L el número de valores distintos que puede tomar la variable objetivo C (número de clases). Se incluyen referencias a otros trabajos en los que se han utilizado estos conjuntos de datos para la selección de variables o clasificación. La columna *Error* es el error cometipo por Naive Bayes con todas las variables usando validación cruzada y 5 *folds*.

4.4. Comparación de las medidas utilizadas

El principal objetivo de los experimentos realizados es probar el rendimiento de las distintas medidas de similitud propuestas con el algoritmo DQPFS. Como objetivo secundario se tiene la comparación de QPFS y DQPFS utilizando la Información Mutua y la Correlación de Pearson. Por simplicidad, se ha preferido incorporar para cada conjunto de datos una gráfica que compara el rendimiento de aplicar QPFS y DQPFS con las distintas medidas de similitud con los mejores α y β seleccionados en cada caso. Pueden encontrarse los resultados completos para cada conjunto de datos en el Anexo A.

Conjunto de datos MUSK.

La Figura 4.5 presenta el porcentaje promedio de acierto en clasificación (Naive Bayes) para el problema MUSK en función del tamaño del subconjunto de variables seleccionado por cada método y las distintas medidas de similitud. Si se desea ver los resultados completos, que incluyen las variaciones de α y β, consúltense en el Anexo A. En la Tabla 4.3 se muestra la tasa de acierto alcanzado por Naive Bayes en el conjunto de *test* y variando el porcentaje de variables seleccionadas.

Para este conjunto de datos todos los métodos y medidas de similitud cosiguen mejorar la clasificación realizada por Naive Bayes con todas las variables en casi un 10 %, a excepción

de la Distancia de Covarianzas. En todos los casos también se consigue mejorar el acierto en clasificación aplicando los métodos propuestos en el conjunto de test. Las medidas que consiguen alcanzar el 80 % de acierto son la Información Mutua Condicionada, la Distancia de Correlaciones y la Información Mutua, pero todas ellas son mejores que la selección aleatoria de variables. La elección de $\beta = 0.75$ indica que es mejor tener en cuenta la redundancia entre variables dada la clase. Además, la Información Mutua Condicionada es el método que mejor funciona a partir de 40 variables seleccionadas. Todas las medidas consiguen igualar o superar el error base del conjunto de test seleccionando el 25 % del total de variables (166). Tanto el error en validación como en test son muy cercanos, lo que hace suponer que no hay *overfitting* en el conjunto de entrenamiento.

Figura 4.5: Acierto en clasificación para el conjunto MUSK en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento.

Método	**Parámetros**	**5 %**	**10 %**	**25 %**	**50 %**	**100 %**
DQPFS + CMI	$\alpha = \hat{\hat{\alpha}}$, $\beta = 0.75$	0.688	0.771	0.771	0.802	0.750
DQPFS + DCOR	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.604	0.677	0.802	0.812	0.750
DQPFS + DCOV	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.656	0.667	0.740	**0.823**	0.750
DQPFS + MI	$\alpha = 0.75$, $\beta = NP$	0.688	0.771	0.792	0.802	0.750
DQPFS + PEARSON	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.604	0.740	0.812	0.802	0.750
QPFS + MI	$\alpha = 0.75$, $\beta = NP$	0.719	0.740	0.771	0.781	0.750
QPFS + PEARSON	$\alpha = 0.5$, $\beta = NP$	0.667	0.740	0.771	0.802	0.750

Tabla 4.3: Acierto en clasificación en el conjunto de test para el conjunto de datos MUSK, donde las columnas de la forma K % indican el procentaje seleccionado sobre el total de variables M, $\alpha = \hat{\hat{\alpha}}$ indica el uso del α empírico, y $\beta = NP$ indica que no procede el uso del parámetro β en la medida de similitud.

Conjunto de datos DIGITS.

La Figura 4.6 presenta el porcentaje promedio de acierto en clasificación (Naive Bayes) para el problema DIGITS en función del tamaño del subconjunto de variables seleccionado por cada

método y las distintas medidas de similitud. Si se desea ver los resultados completos, que incluyen las variaciones de α y β, consúltense en el Anexo A. En la Tabla 4.4 se muestra la tasa de acierto alcanzado por Naive Bayes en el conjunto de *test* y variando el porcentaje de variables seleccionadas.

En este conjunto todos los métodos de selección superan, cuando se escogen menos de 100 variables, a la selección aleatoria de variables. A partir de 100 variables la Distancia de Covarianzas comienza a tener peores resultados, llegando incluso a estar por debajo de la selección aleatoria de variables. Tanto QPFS y DQPFS con la Correlación de Pearson, la Información Mutua y la Información Mutua Condicionada consiguen mejorar la tasa de acierto del clasificador con todas las variables hasta llegar cerca del 80 % de acierto. Tienen tasas peores las Distancias de Covarianzas y Correlaciones. En el conjunto de test, que partía de un acierto base del 60.1 % correspondiente de clasificar con todas las variables, se alcanza el máximo acierto del 77.01 % seleccionando un 25 % de las variables del total de 190.

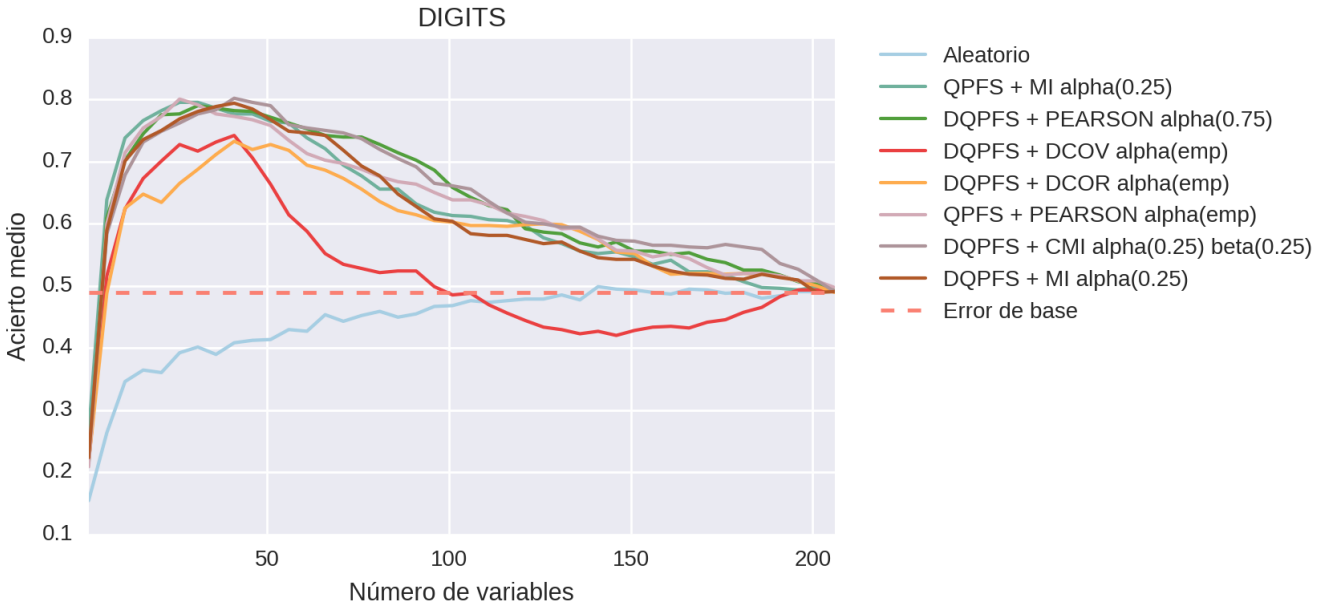


Figura 4.6: Acierto en clasificación para el conjunto DIGITS en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento.

Método	**Parámetros**	**5 %**	**10 %**	**25 %**	**50 %**	**100 %**
DQPFS + CMI	$\alpha = 0.25$, $\beta = 0.25$	0.665	0.734	**0.771**	0.644	0.601
DQPFS + DCOR	$\alpha = \hat{\alpha}$, $\beta = NP$	0.574	0.622	0.713	0.580	0.601
DQPFS + DCOV	$\alpha = \hat{\alpha}$, $\beta = NP$	0.574	0.644	0.691	0.564	0.601
DQPFS + MI	$\alpha = 0.25$, $\beta = NP$	0.654	0.734	0.766	0.601	0.601
DQPFS + PEARSON	$\alpha = 0.75$, $\beta = NP$	0.633	0.734	**0.771**	0.697	0.601
QPFS + MI	$\alpha = 0.25$, $\beta = NP$	0.707	0.723	**0.771**	0.596	0.601
QPFS + PEARSON	$\alpha = \hat{\alpha}$, $\beta = NP$	0.628	**0.771**	0.707	0.660	0.601

Tabla 4.4: Acierto en clasificación en el conjunto de test para el conjunto de datos DIGITS, donde las columnas de la forma K % indican el procentaje seleccionado sobre el total de variables M, $\alpha = \hat{\alpha}$ indica el uso del α empírico, y $\beta = NP$ indica que no procede el uso del parámetro β en la medida de similitud.

Conjunto de datos LUNG.

La Figura 4.7 presenta el porcentaje promedio de acierto en clasificación (Naive Bayes) para el problema LUNG en función del tamaño del subconjunto de variables seleccionado por cada método y las distintas medidas de similitud. En la Tabla 4.5 está la tasa de acierto alcanzado por Naive Bayes en el conjunto de *test* y variando el porcentaje de variables seleccionadas.

En este conjunto, aunque la mayoría de los métodos son mejores que la selección aleatoria, se observa que seleccionar variables aleatoriamente mejora la clasificación rápidamente, lo que nos hace pensar que casi todas las variables del problema son relevantes y aportan información útil. El método que mejor tasa de acierto consigue, superando el 90 % para unas 100 variables es DQPFS con la Información Mutua Condicionada. El valor óptimo de $\beta = 1$ para DQPFS con la Información Mutua Condicionada confirma que es útil tener en cuenta la redundancia entre variables dada la clase. En el conjunto de test todos los métodos y medidas consiguen alcanzar el error base seleccionando el 50 % de las variables.

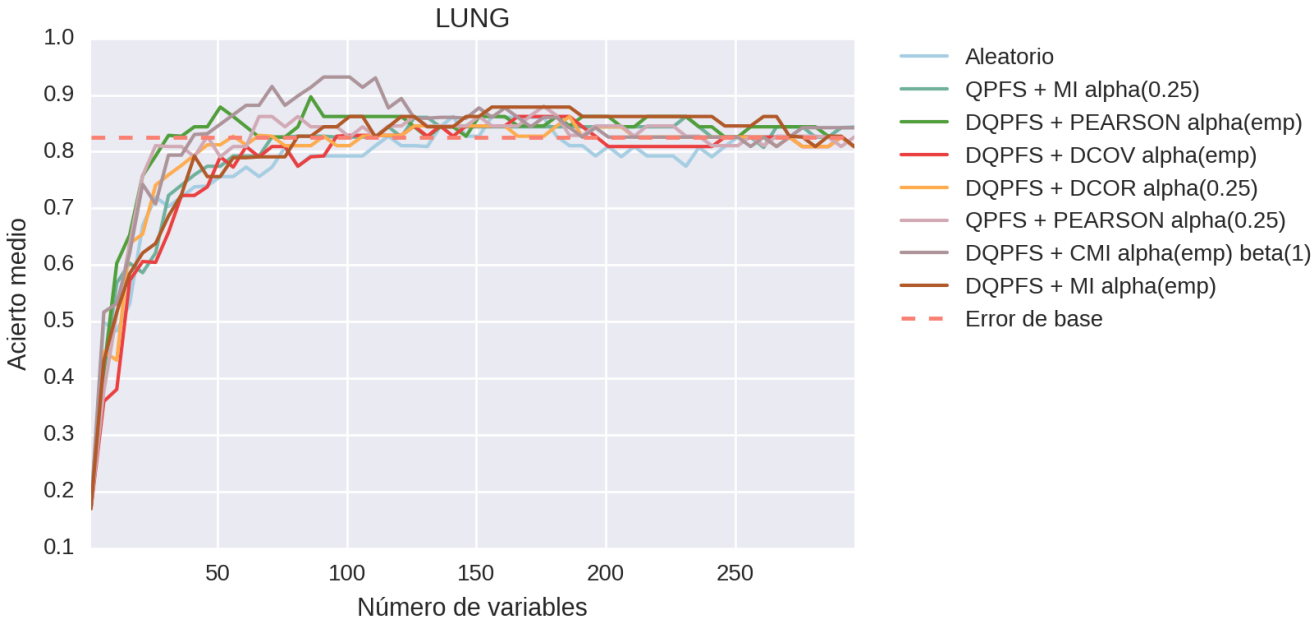


Figura 4.7: Acierto en clasificación para el conjunto LUNG en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento.

Método	**Parámetros**	**5 %**	**10 %**	**25 %**	**50 %**	**100 %**
DQPFS + CMI	$\alpha = \hat{\alpha}$, $\beta = 1$	0.600	0.867	0.867	0.933	0.933
DQPFS + DCOR	$\alpha = 0.25$, $\beta = NP$	0.800	0.800	0.867	0.933	0.933
DQPFS + DCOV	$\alpha = \hat{\alpha}$, $\beta = NP$	0.667	0.867	0.800	0.933	0.933
DQPFS + MI	$\alpha = \hat{\alpha}$, $\beta = NP$	0.667	0.867	0.933	0.933	0.933
DQPFS + PEARSON	$\alpha = \hat{\alpha}$, $\beta = NP$	0.733	0.867	0.867	0.933	0.933
QPFS + MI	$\alpha = 0.25$, $\beta = NP$	0.933	**1.000**	**1.000**	0.933	0.933
QPFS + PEARSON	$\alpha = 0.25$, $\beta = NP$	0.733	0.933	0.933	0.933	0.933

Tabla 4.5: Acierto en clasificación en el conjunto de test para el conjunto de datos LUNG, donde las columnas de la forma K % indican el procentaje seleccionado sobre el total de variables M, $\alpha = \hat{\alpha}$ indica el uso del α empírico, y $\beta = NP$ indica que no procede el uso del parámetro β en la medida de similitud.

Conjunto de datos WDBC.

La Figura 4.8 presenta el porcentaje promedio de acierto en clasificación (Naive Bayes) para el problema WDBC en función del tamaño del subconjunto de variables seleccionado por cada método y las distintas medidas de similitud. Si se desea ver los resultados completos, que incluyen las variaciones de α y β, pueden verse en el Anexo A. En la Tabla 4.6 se muestra la tasa de acierto alcanzado por Naive Bayes en el conjunto de *test* y variando el porcentaje de variables seleccionadas.

Los resultados en este conjunto de datos son similares para todos los métodos y medidas. Todos consiguen mejorar a la selección aleatoria de variables, pero no se puede elegir un ganador. El método con la menor mejora respecto al acierto base es DQPFS con la Correlación de Pearson. En el conjunto de test, que partía con un acierto inicial al usar todas las variables del 97.4 %, ningún método de selección consigue mejorar al clasificar con todas las variables, pero todos consiguen tasas similares seleccionando solo el 25 % de las variables.

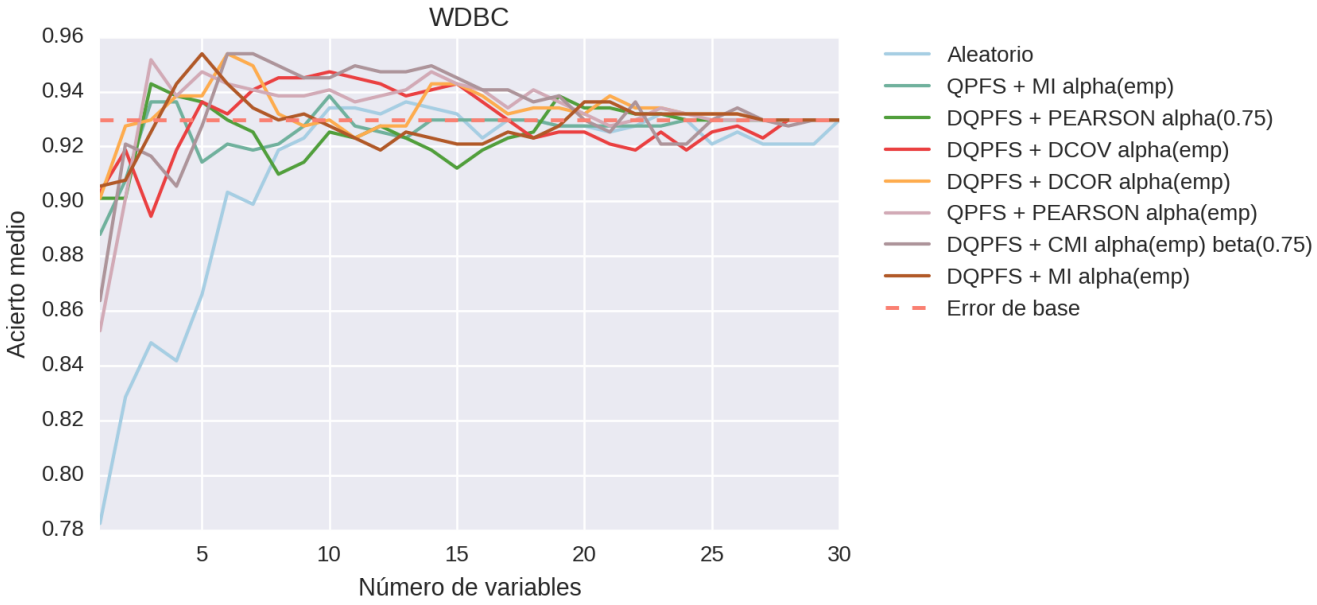


Figura 4.8: Acierto en clasificación para el conjunto WDBC en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento.

Método	**Parámetros**	**5 %**	**10 %**	**25 %**	**50 %**	**100 %**
DQPFS + CMI	$\alpha = \hat{\hat{\alpha}}$, $\beta = 0.75$	0.921	0.912	0.930	0.965	0.974
DQPFS + DCOR	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.921	0.965	**0.974**	**0.974**	0.974
DQPFS + DCOV	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.947	0.886	0.939	**0.947**	0.974
DQPFS + MI	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.921	0.904	0.947	0.965	0.974
DQPFS + PEARSON	$\alpha = 0.75$, $\beta = NP$	0.921	0.965	**0.974**	**0.974**	0.974
QPFS + MI	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.921	0.921	0.956	0.965	0.974
QPFS + PEARSON	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.921	0.956	0.956	**0.974**	0.974

Tabla 4.6: Acierto en clasificación en el conjunto de test para el conjunto de datos WDBC, donde las columnas de la forma $K\,\%$ indican el procentaje seleccionado sobre el total de variables M, $\alpha = \hat{\hat{\alpha}}$ indica el uso del α empírico, y $\beta = NP$ indica que no procede el uso del parámetro β en la medida de similitud.

Conjunto de datos WINE.

La Figura 4.9 presenta el porcentaje promedio de acierto en clasificación (Naive Bayes) para el problema WINE en función del tamaño del subconjunto de variables seleccionado por cada método y las distintas medidas de similitud. Los resultados completos, que incluyen las variaciones de α y β, se encuentran en el Anexo A. En la Tabla 4.7 se muestra el acierto alcanzado por Naive Bayes en el conjunto de *test* y variando el porcentaje de variables seleccionadas.

En este sencillo conjunto de datos todos los métodos de selección consiguen superar a la selección aleatoria de variables y obtienen un rendimiento muy similar entre sí, a excepción de DQPFS con la Distancia de Covarianzas, que consigue un rendimiento bastante inferior al resto de medidas. Ocurre lo mismo en el conjunto de *test*, donde todos los métodos consiguen igualar el error en clasificación con todas las variables utilizando sólo un 50 % de las mismas, excepto la Distancia de Covarianzas.

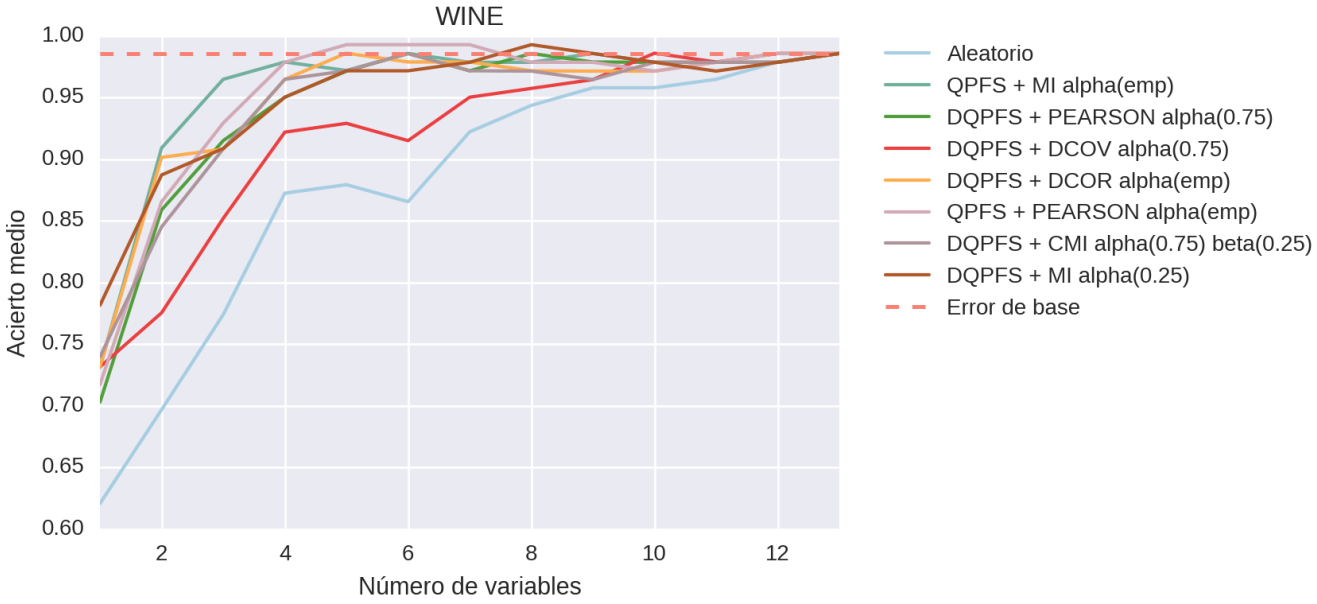


Figura 4.9: Acierto en clasificación para el conjunto WINE en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento.

Método	**Parámetros**	**10 %**	**25 %**	**50 %**	**100 %**
DQPFS + CMI	$\alpha = 0.75$, $\beta = 0.25$	0.722	0.861	**0.917**	0.917
DQPFS + DCOR	$\alpha = \hat{\alpha}$, $\beta = NP$	0.667	0.889	**0.917**	0.917
DQPFS + DCOV	$\alpha = 0.75$, $\beta = NP$	0.667	0.611	0.861	0.917
DQPFS + MI	$\alpha = 0.25$, $\beta = NP$	0.611	0.806	**0.917**	0.917
DQPFS + PEARSON	$\alpha = 0.75$, $\beta = NP$	0.667	0.806	**0.917**	0.917
QPFS + MI	$\alpha = \hat{\alpha}$, $\beta = NP$	0.722	0.889	**0.917**	0.917
QPFS + PEARSON	$\alpha = \hat{\alpha}$, $\beta = NP$	0.667	0.806	**0.917**	0.917

Tabla 4.7: Acierto en clasificación en el conjunto de test para el conjunto de datos WINE, donde las columnas de la forma K % indican el procentaje seleccionado sobre el total de variables M, $\alpha = \hat{\alpha}$ indica el uso del α empírico, y $\beta = NP$ indica que no procede el uso del parámetro β en la medida de similitud.

Conjunto de datos AUDIOLOGY.

La Figura 4.10 presenta el porcentaje promedio de acierto en clasificación (Naive Bayes) para el problema AUDIOLOGY en función del tamaño del subconjunto de variables seleccionado por cada método y las distintas medidas de similitud. Los resultados completos, que incluyen las variaciones de α y β, pueden verse en el Anexo A. En la Tabla 4.8 se muestra la tasa de acierto alcanzado por Naive Bayes en el conjunto de *test* y variando el porcentaje de variables seleccionadas.

En este conjunto de datos el método que mejores resultados ha obtenido es DQPFS con la Información Mutua y eligiendo el parámetro α heurístico, por lo que es un buen ejemplo para observar que no siempre conviene penalizar a las variables más entrópicas. Es el único método que, seleccionando variables, consigue que Naive Bayes alcance el 60 % de acierto en clasificación. También es el método que mejor rendimiento consigue en el conjunto de *test* al seleccionar el 50 % de las variables, junto a DQPFS con la Información Mutua Condicionada.

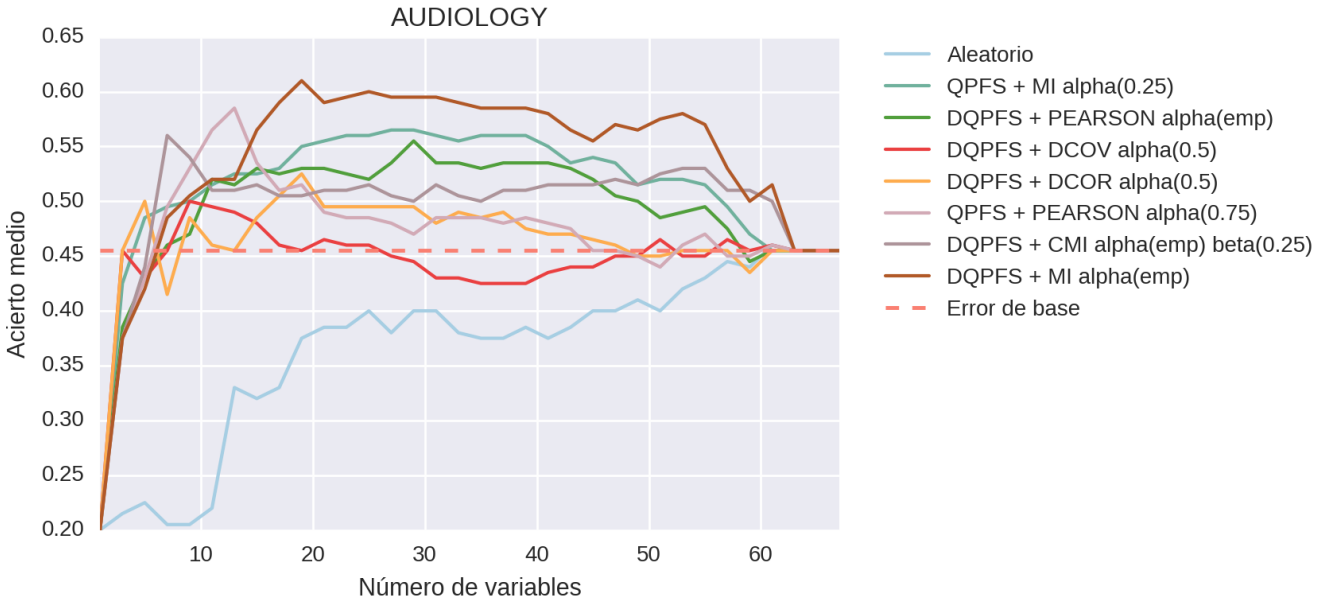


Figura 4.10: Acierto en clasificación para el conjunto AUDIOLOGY en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento.

Método	**Parámetros**	**5 %**	**10 %**	**25 %**	**50 %**	**100 %**
DQPFS + CMI	$\alpha = \hat{\alpha}$, $\beta = 0.25$	0.423	0.577	0.577	0.654	0.615
DQPFS + DCOR	$\alpha = 0.5$, $\beta = NP$	0.423	0.462	0.654	0.615	0.615
DQPFS + DCOV	$\alpha = 0.5$, $\beta = NP$	0.423	0.654	0.500	0.615	0.615
DQPFS + MI	$\alpha = \hat{\alpha}$, $\beta = NP$	0.423	0.500	**0.692**	0.654	0.615
DQPFS + PEARSON	$\alpha = \hat{\alpha}$, $\beta = NP$	0.538	0.538	0.615	0.577	0.615
QPFS + MI	$\alpha = 0.25$, $\beta = NP$	0.423	0.462	0.577	0.615	0.615
QPFS + PEARSON	$\alpha = 0.75$, $\beta = NP$	0.423	0.538	0.615	0.615	0.615

Tabla 4.8: Acierto en clasificación en el conjunto de test para el conjunto de datos AUDIOLOGY, donde las columnas de la forma K % indican el procentaje seleccionado sobre el total de variables M, $\alpha = \hat{\alpha}$ indica el uso del α empírico, y $\beta = NP$ indica que no procede el uso del parámetro β en la medida de similitud.

Conjunto de datos URBAN.

La Figura 4.11 presenta el porcentaje promedio de acierto en clasificación (Naive Bayes) para el problema URBAN en función del tamaño del subconjunto de variables seleccionado por cada método y las distintas medidas de similitud. Los resultados completos, que incluyen las variaciones de α y β, se encuentran en el Anexo A. En la Tabla 4.9 se muestra la tasa de acierto alcanzado por Naive Bayes en el conjunto de *test* y variando el porcentaje de variables seleccionadas.

En este conjunto se obtienen resultados similares (o incluso mejores) al entrenar el clasificador con una selección aleatoria de las variables que al aplicar un método de selección. Los métodos que consiguen una mejora significativa con respecto a la selección aleatoria, incluso ligeramente superiores al acierto base, son QPFS (seleccionando aproximadamente 10 variables) con la Información Mutua y la Correlación de Pearson y DQPFS (seleccionando aproximadamente 20 variables) con la Información Mutua Condicionada.

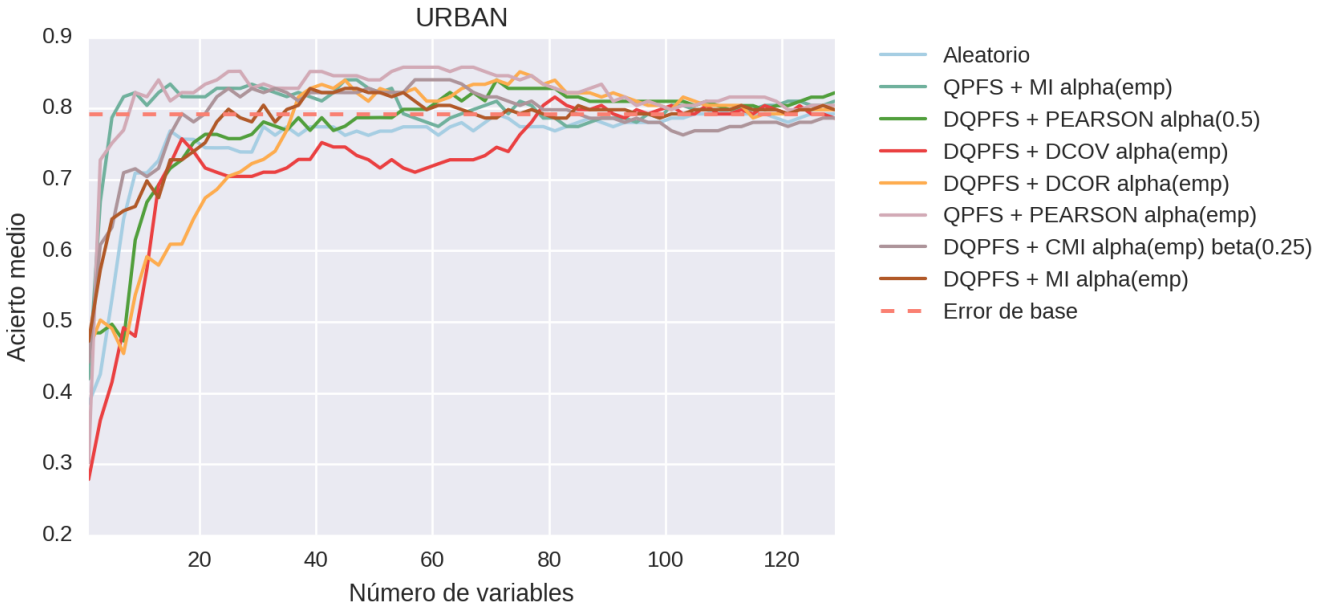


Figura 4.11: Acierto en clasificación para el conjunto URBAN en función del número de variables seleccionadas y utilizando α y β con el mejor rendimiento en el conjunto de entrenamiento.

Método	**Parámetros**	**5 %**	**10 %**	**25 %**	**50 %**	**100 %**
DQPFS + CMI	$\alpha = \hat{\hat{\alpha}}$, $\beta = 0.25$	0.706	0.710	0.755	**0.793**	0.767
DQPFS + DCOR	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.521	0.564	0.738	0.748	0.767
DQPFS + DCOV	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.422	0.641	0.667	0.700	0.767
DQPFS + MI	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.706	0.728	0.755	0.757	0.767
DQPFS + PEARSON	$\alpha = 0.5$, $\beta = NP$	0.521	0.694	0.724	0.769	0.767
QPFS + MI	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.751	0.761	0.781	**0.791**	0.767
QPFS + PEARSON	$\alpha = \hat{\hat{\alpha}}$, $\beta = NP$	0.767	0.795	0.789	**0.801**	0.767

Tabla 4.9: Acierto en clasificación en el conjunto de test para el conjunto de datos URBAN, donde las columnas de la forma $K\,\%$ indican el procentaje seleccionado sobre el total de variables M, $\alpha = \hat{\hat{\alpha}}$ indica el uso del α empírico, y $\beta = NP$ indica que no procede el uso del parámetro β en la medida de similitud.

4.5. Complejidad temporal de las medidas de información

En la Sección 3.4 se hizo un análisis teórico del coste computacional del cálculo de las distintas medidas de información entre variables aleatorias que se introducen en este trabajo. Para corroborar estos resultados, se han hecho algunos experimentos con ejemplos sintéticos. En las Figuras 4.12 y 4.13 se muestran los tiempos de ejecución del cálculo de la Distancia de Covarianzas y de Correlaciones, la Información Mutua y la Información Mutua Condicionada para dos variables aleatorias donde N representa el número de muestras. Para cada N fijo y cada medida de similitud se han realizado 10 ejecuciones y se ha obtenido la media del tiempo de cálculo en cada una de estas configuraciones. Para la Información Mutua Condicionada se ha realizado el análisis del coste computacional considerando una tercera variable aleatoria que puede tomar dos valores distintos para simular la clase (sería un ejemplo con $L = 2$).

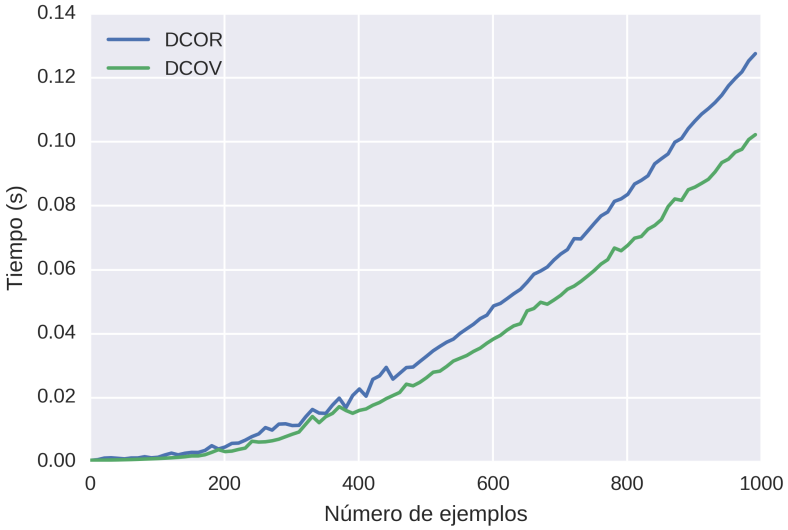


Figura 4.12: Tiempos medios de ejecución en segundos en 10 iteraciones del cálculo de la Distancia de Covarianzas y Correlaciones variando el número de muestras de un conjunto de datos sintético.

Como resultado, para la Distancia de Covarianzas y de Correlaciones se observa un rendimiento cuadrático en el número de ejemplos (Figura 4.12) y un rendimiento lineal para las medidas basadas en Información Mutua (Figura 4.13), tal y como se analizó teóricamente en la Sección 3.4. Existe una implementación más rápida de la Distancia de Covarianzas y la Distancia de Correlaciones que se desarrolló en el Grupo de Aprendizaje Automático de la *Universidad Autónoma de Madrid* y basada en [44]. Por motivos de tiempo no ha podido incluirse en este análisis.

En la Figura 4.14 se muestra un experimento variando el número de clases distintas del conjunto de datos sintético. El procedimiento seguido es el siguiente: para un número de muestras fijo $N = 600$ se ha variado entre 1 y 50 el número de posibles clases distintas. Esto se consigue variando el número de valores distintos que puede tomar la variable que simula a la clase en el cálculo de la Información Mutua Condicionada y generando otras dos variables aleatorias de 600 muestras. El aumento del número de clases no afecta a las Distancias de Covarianzas y Correlaciones ni a la Información Mutua, ya que su cálculo es independiente del número de clases distintas que tenga el problema. En el caso de la Información Mutua Condicionada, el tiempo de ejecución crece linealmente respecto al número de clases posibles, como ya se mostró en la Sección 3.4.

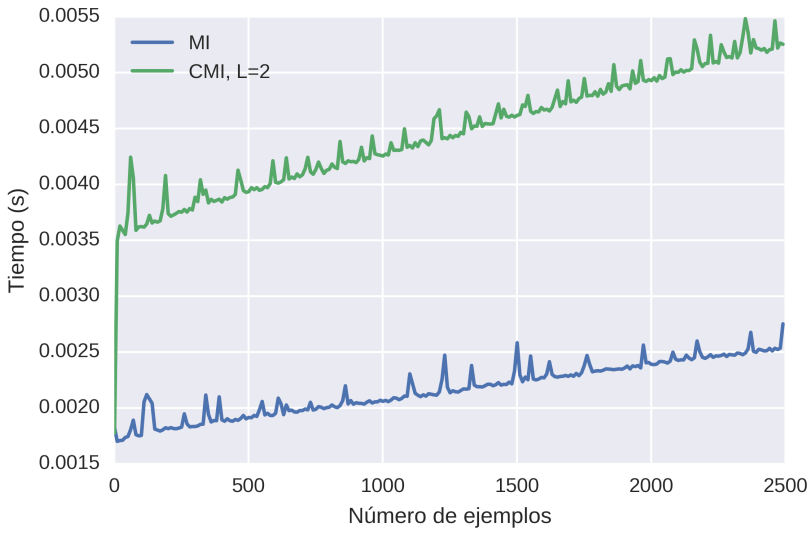


Figura 4.13: Tiempos medios de ejecución en segundos en 10 iteraciones del cálculo de la Información Mutua y la Información Mutua Condicionada variando el número de muestras de un ejemplo sintético.

Por último, en la Figura 4.15 se muestran los tiempos de ejecución de las distintas medidas de información variando, de nuevo, el número de ejemplos N. Esta vez hacemos N variar en un intervalo más pequeño, siendo $N = 250$ el número máximo de muestras que vamos a cosiderar. En este caso, la variable aleatoria que simula a la clase puede tomar 2 valores distintos. La finalidad de este experimento es mostrar que, si bien el coste teórico de ejecución de la Distancia de Covarianzas y Correlaciones es cuadrático respecto al número de ejemplos, y, por lo tanto, según crece el número de ejemplos aumenta mucho más rápido que el de la Información Mutua o la Información Mutua Condicionada (cuyo rendimiento es lineal respecto a N), para un número pequeño de variables resultan más rápidas las dos primeras medidas. Esto se debe a que en el caso de la Información Mutua y la Información Mutua Condicionada aparece una cota inferior en el tiempo de ejecución que no aparece en el caso de las Distancias de Covarianzas y Correlaciones. Esta cota puede aparecer por cuestiones internas de las librerías utilizadas o el tiempo empleado en la discretización de los datos y no es objeto de este TFG.

En la Figura 4.16 se ha representado el coste computacional promedio en 5 iteraciones de los algoritmos QPFS y DQPFS en los conjuntos de datos del mundo real que se han utilizado en los experimentos de la Sección 4.4 y se describen en la Tabla 4.2. Se observa que, en ocasiones, el tiempo de ejecución de realizar la selección de variables usando la Distancia de Covarianzas y Correlaciones es menor que al usar la Información Mutua, tal y como se había comprobado anteriormente.

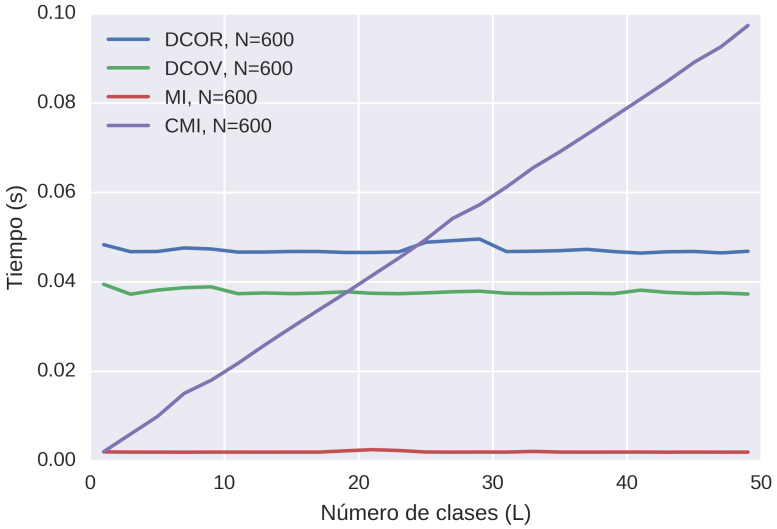


Figura 4.14: Tiempos de ejecución en segundos 10 iteraciones del cálculo de la Información Mutua, la Información Mutua Condicionada y la Distancia de Covarianzas y Correlaciones variando el número de clases distintas de un conjunto de datos sintético.

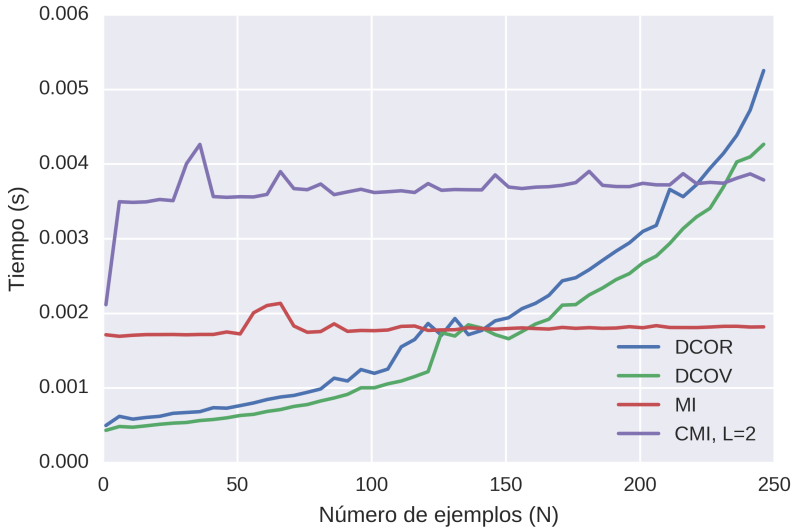
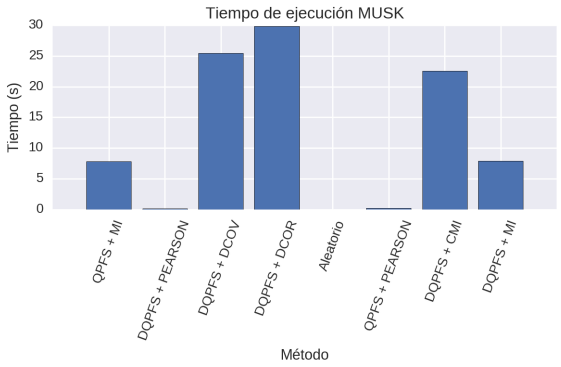
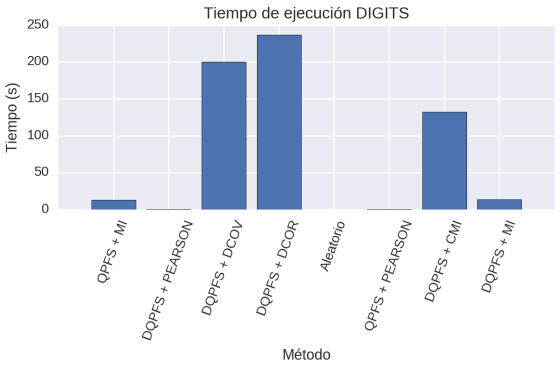


Figura 4.15: Tiempos medios de ejecución en segundos 10 iteraciones del cálculo de Información Mutua, la Información Mutua Condicionada y la Distancia de Covarianzas y Correlaciones variando el número de muestras de un conjunto de datos sintético.

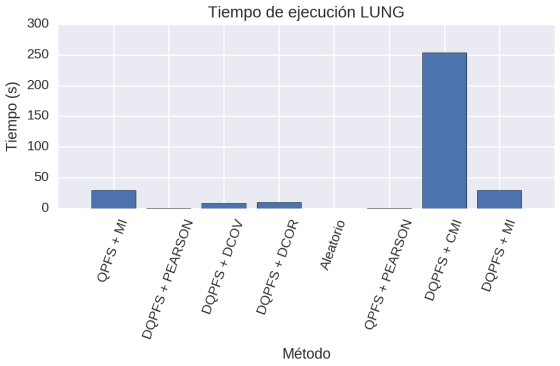


(a) MUSK

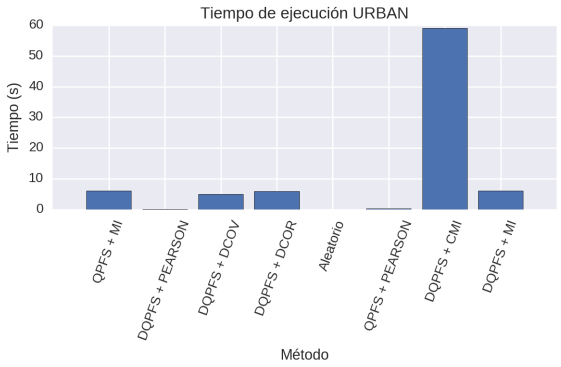


(b) DIGITS

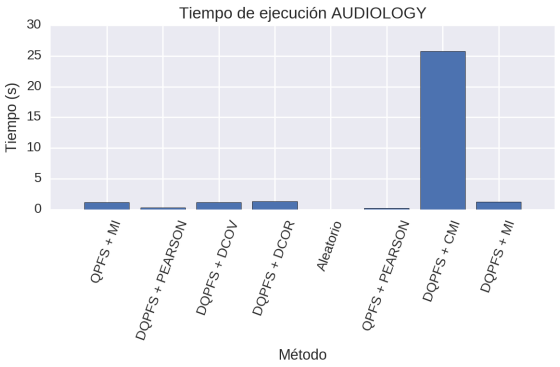
(c) WINE



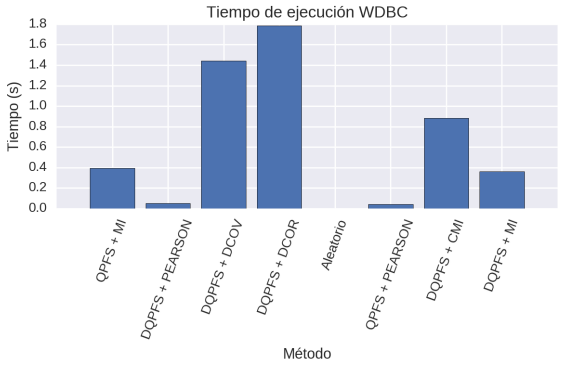
(d) LUNG



(e) URBAN



(f) AUDIOLOGY



(g) WDBC

Figura 4.16: Tiempos medios de ejecución del algoritmo de selección de variables en 5 iteraciones.

5

Conclusiones y trabajo futuro

5.1. Conclusiones

El objetivo de este TFG era analizar el rendimiento de distintas medidas de similitud entre variables alatorias aplicadas a un método global de selección de variables. Realizar el trabajo ha llevado un gran esfuerzo de lectura y comprensión del paradigma de selección de variables, los métodos existentes y las distintas medidas de similitud previamente empleadas en la literatura. El desarrollo se ha hecho en tres etapdas diferentes. Primero, una etapa de adquisición de los conocimientos necesarios para la comprensión del objetivo del TFG. Una segunda etapa de programación, centrada en la implementación y prueba de QPFS, DQPFS y las medidas de similitud así como las pruebas posteriores para verificar su correcto funcionamiento. Por último, la fase de realización de experimentos, en la que se eligieron los conjuntos de datos a utilizar para el posterior análisis de los resultados. Todo el proyecto se ha realizado en *Python*, incluyendo la representación de resultados.

La mayoría de las medidas de similitud propuestas en este trabajo han tenido un rendimiento positivo al aplicarse a un método global de selección de variables, consiguiendo mejorar el acierto en clasificación alcanzado por el clasificador Naive Bayes usando todas las variables o alcanzando la misma tasa de acierto con menos variables y, por tanto, simplificando el problema y el coste de entrenar un clasificador. No obstante, en algunas ocasiones, realizar una simple selección aleatoria de variables puede conseguir buenos resultados (ver Figuras A.3 y A.7) y, en otros casos, algunas de las medidas no consiguen resultados demasiado prometedores (ver Figura A.2).

Merece la pena destacar que en muchos de los conjuntos de datos, para cada método, se ha conseguido el mejor rendimiento utilizando el parámetro α empírico explicado en la Sección 3.2, lo que indica que esta elección de α es acertada y consigue equiparar los pesos que tienen la redundancia entre variables y la relevancia en los problemas de selección de variables.

A pesar del éxito conseguido por la Distancia de Covarianzas y la Distancia de Correlaciones en algunos métodos de selección de variables iterativos como mRMR [18], estas métricas no han conseguido superar al rendimiento que consigue QPFS en su implementación original [1] en casi ninguno de los conjuntos de datos utilizados en los experimentos de este TFG, siendo, además, medidas más costosas de calcular que la Información Mutua y la Correlación de Pearson. En el Anexo A se puede ver que la Distancia de Covarianzas funciona, en algunas ocasiones,

notablemente peor que una selección aleatoria de las variables en algunos de los conjuntos de datos propuestos. Por otro lado, la Distancia de Correlaciones sí consigue superar a la selección aleatoria de variables (ver Figuras A.2, A.4). Esto se debe a que la Distancia de Correlaciones es una medida acotada entre 0 y 1, pero la Distancia de Covarianzas puede tomar valores arbitrariamente grandes. Esto crea un desequilibrio entre las maginitudes de la matriz Q y el vector F que sólo consigue solucionarse con el uso del α empírico, $\hat{\hat{\alpha}}$.

Por otra parte, el uso las variables seleccionadas al aplicar DQPFS con la Información Mutua Condicionada sí ha supuesto una mejora en el acierto en clasificación de algunos de los conjuntos de datos (MUSK, LUNG y el ejemplo sintético de la Sección 4.2) respecto a aplicar QPFS o DQPFS con la Información Mutua o la Correlación de Pearson. Aplicar DQPFS con esta medida de similitud tiene el inconveniente de la aparición del parámetro β, y, por tanto, su elección o estimación. Además, es ligeramente más costoso que DQPFS con la Información Mutua o la Correlación de Pearson, ya que el coste computacional del cálculo de la Información Mutua Condicionada, aunque es lineal en el número de patrones, también aumenta linealmente respecto al número de clases distintas. En cualquier caso, el uso de esta medida de información resulta bastante prometedor, pudiendo aliviarse el coste computacional con el uso de aproximaciones como el Método de Nyström, que ya se utilizó en la implementación original de QPFS [1].

La introdución de la constante γ en la diagonal de la matriz Q, si bien ha demostrado una mejora del acierto en clasificación en algunos de los conjuntos de datos empleados (por ejemplo, en AUDIOLOGY), no ha supuesto grandes mejoras respecto a la implementación original de QPFS. En cualquier caso, tiene la ventaja de que garantiza estar ante un problema convexo en el espacio original de las variables y resuelve el problema de QPFS planteado en [9], donde se prueba que el uso original de la entropia en la diagonal de Q puede dirigir a soluciones subóptimas. Por todo ello, se recomienda el uso del algoritmo DQPFS.

Durante el transcurso de este Trabajo de Fin de Grado se ha puesto en práctica gran parte del conocimiento adquirido durante la carrera, tanto en el Grado en Matemáticas como en el Grado en Ingeniería Informática. Para la parte de implementación, una de las asignaturas que más ha influido es *Fundamentos de Aprendizaje Automático*, cuyas prácticas se realizaron en *Python*, el mismo lenguaje en el que se ha desarrollado este trabajo. Además, ha resultado útil el material desarrollado durante las práticas de esta asignatura para leer y procesar conjuntos de datos, así como para la representación final de los resultados. En lo referente al Grado en Matemáticas, a parte del conocimiento necesario de *Estadística* para entender y analizar las medidas de similitud, también resulta útil *Investigación Operativa* a la hora de resolver problemas de optimización. Los conocimientos adquiridos en la asignatura de *Algebra Lineal*, que trata las propiedades básicas de las matrices, sus autovalores y sus autovectores, también han sido clave en para la correcta comprensión del trabajo. Además, dada que toda la información de este TFG procedía de artículos científicos, el estar acostumbrado a este tipo de lectura ha sido de gran ayuda.

5.2. Trabajo futuro

El algoritmo de selección de variables DQPFS y las medidas de similitud empleadas dejan abiertas algunas líneas de investigación para sucesivas mejoras o aplicaciones a distintos campos del aprendizaje automático. A continuación se describen brevemente algunas de las principales direcciones en las que puede ir orientada la investigación basada en los conceptos desarrollados y resultados obtenidos a lo largo de este TFG.

Discretización de los datos. Elegir un método de discretización de las variables continuas para el cálculo de la Información Mutua y la Información Mutua condicionada no es una tarea

trivial. Este trabajo se ha centrado en la discretización en tres intervalos marcados por la media y la desviación típica de la variable, tal y como se propone en el algoritmo original de QPFS [1]. No obstante, utilizar otros métodos de discretización puede tener un gran impacto en el ranking final generado por el algoritmo de selección, tal y como se muestra en la Sección 4.2. Un trabajo futuro sería probar DQPFS con la Información Mutua Condicionada y la Información Mutua con otros métodos de discretización, como la discretización de *Doane*, explicada en la Sección 4.1.

Implementación de una librería. El algoritmo QPFS original está implementado en C y en Matlab y puede descargar de `https://sites.google.com/site/irenerodriguezlujan/documents/qpfs`. No obstante, no hay una librería implementada en *Python*. Un trabajo futuro es adaptar el código producido en este TFG para ajustarse a los estándares de *SCIKIT-LEARN*, la librería más utilizada de Aprendizaje Automático en *Python*, y publicarlo para su uso.

Pruebas con más clasificadores. Aunque Naive Bayes resulta un clasificador muy atractivo a la hora de probar métodos de selección de variables que buscan minimizar la redundancia (mRMR, QPFS) ya que parte de la hipótesis de que las variables del problema son independientes entre sí, podría resultar interesante probar todos los métodos y las medidas propuestas sobre otro clasificador, como, por ejemplo Máquinas de Vectores Soporte [14].

Selección de los hiperparámetros. Tanto QPFS como DQPFS requieren definir los parámetros α y β. En este trabajo, la búsquda de los parámetros α y β adecuados se ha hecho mediante búsqueda en rejilla, por lo que no se puede garantizar que los parámetros elegidos sean óptimos. Se deja abierto como trabajo futuro el estudio de métodos que permitan elegir inteligentemente estos parámetros.

Método de Nyström. Por restricciones de tiempo, no ha sido posible implementar QPFS y DQPFS con el método de Nyström [1]. El método de Nyström permite calcular los autovalores y autovectores de Q resolviendo un problema de diagonalización en un espacio de menor dimensión a partir un submuestreo de las filas de la matriz del término cuadrático, lo que supone un descenso vertiginoso en el tiempo de ejecución que resulta muy conveniente debido al coste computacional del cálculo de la mayoría de las medidas de información entre variables aleatorias. Resultaría interesante realizar pruebas para comparar el redimiento de DQPFS y las medidas de similitud empleadas en este trabajo utilizando esta aproximación.

Determinar el número óptimo de variables. Uno de los principales problemas de los métodos de filtro de selección de variables es que no tienen una forma propia de discriminar qué cantidad de variables tiene el conjunto final con el que vamos a entrenar a nuestro clasificador. Tanto QPFS como DQPFS tienden a asignar un peso arbitrariamente cercano a cero a algunas de las variables del problema, lo que indicaría que no son relevantes. Una línea abierta de investigación es comprobar si este comportamiento sigue algún tipo de patrón para evitar el uso de un clasificador a la hora de seleccionar el número óptimo de variables.

Probar otros conjuntos de datos. En el artículo original [1] se muestra que QPFS obtiene muy buenos resultados en conjuntos de datos genéticos, que se componen de un número muy elevado de variables pero de pocos ejemplos ($N << M$). Por motivos de coste computacional y escasez de recursos, no se han podido llevar a cabo experimentos en conjuntos de este tipo, a excepción del conjunto LUNG, que no resultaba especialmente interesante porque que la selección

aleatoria ya tenía un rendimiento muy elevado. Por tanto, el análisis del rendimiento de DQPFS y las medidas de similitud implementadas en este TFG sobre conjuntos de datos genéticos es otra línea de trabajo futuro.

Glosario de acrónimos

- **TFG**: Trabajo de Fin de Grado
- **DCOR**: Distancia de Correlaciones
- **DCOV**: Distancia de Covarianzas
- **QP**: Quadratic Programming - Programación cuadrática
- **MI**: Información Mutua
- **CMI**: Información Mutua Condicionada
- **PEARSON**: Coeficiente de Correlación de Pearson
- **QPFS**: Quadratic Programming Feature Selection
- **DQPFS**: Diagonal Quadratic Programming Feature Selection

Bibliografía

[1] Irene Rodriguez-Lujan, Ramon Huerta, Charles Elkan, and Carlos Santa Cruz. Quadratic programming feature selection. *Journal of Machine Learning Research*, 11(Apr):1491–1516, 2010.

[2] Adam C Pocock. *Feature selection via joint likelihood*. PhD thesis, University of Manchester, 2012.

[3] Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[4] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[5] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[6] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

[7] David D Lewis. Feature selection and feature extraction for text categorization. pages 212–217, 1992.

[8] Hanchuan Peng, Fuhui Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on pattern analysis and machine intelligence*, 27(8):1226–1238, 2005.

[9] Xuan Vinh Nguyen, Jeffrey Chan, Simone Romano, and James Bailey. Effective global approaches for mutual information based feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 512–521. ACM, 2014.

[10] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324, 1997.

[11] Glenn De'ath and Katharina E Fabricius. Classification and regression trees: a powerful yet simple technique for ecological data analysis. *Ecology*, 81(11):3178–3192, 2000.

[12] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[13] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, 1999.

[14] R. O. Duda and D. G. Stork. Pattern classification (2nd edition). *Wiley-Interscience*, 2000.

[15] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, page 27(7):379?423, 1948.

[16] Ralf Steuer, Jürgen Kurths, Carsten O Daub, Janko Weise, and Joachim Selbig. The mutual information: detecting and evaluating dependencies between variables. *Bioinformatics*, 18(suppl 2):S231–S240, 2002.

[17] Gábor J Székely, Maria L Rizzo, Nail K Bakirov, et al. Measuring and testing dependence by correlation of distances. *The Annals of Statistics*, 35(6):2769–2794, 2007.

[18] José R Berrendero, Antonio Cuevas, and José L Torrecilla. The mrmr variable selection method: a comparative study for functional data. *Journal of Statistical Computation and Simulation*, 86(5):891–907, 2016.

[19] Maria L. Rizzo and Gábor J. Székely. Energy distance. *Wiley Interdisciplinary Reviews: Computational Statistics*, 8(1):27–38, 2016.

[20] Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13(Jan):27–66, 2012.

[21] Francisco Macedo, M Rosário Oliveira, António Pacheco, and Rui Valadas. A theoretical framework for evaluating forward feature selection methods based on mutual information. *arXiv preprint arXiv:1701.07761*, 2017.

[22] Roberto Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550, 1994.

[23] Cláudia Pascoal, M Rosário Oliveira, António Pacheco, and Rui Valadas. Theoretical evaluation of feature selection methods based on mutual information. volume 226, pages 168–181. Elsevier, 2017.

[24] Dahua Lin and Xiaoou Tang. Conditional infomax learning: an integrated framework for feature extraction and fusion. *Computer Vision–ECCV 2006*, pages 68–82, 2006.

[25] Howard Hua Yang and John E Moody. Data visualization and feature selection: New algorithms for nongaussian data. In *NIPS*, volume 12, 1999.

[26] François Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5(Nov):1531–1555, 2004.

[27] Mohamed Bennasar, Yulia Hicks, and Rossitza Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

[28] Aleks Jakulin. *Machine learning based on attribute interactions*. PhD thesis, Univerza v Ljubljani, 2005.

[29] D Andersen, J Dahl, and L Vandenberghe. Cvxopt: Python software for convex optimization, 2013.

[30] Mark A Hall. Correlation-based feature selection of discrete and numeric class machine learning. 2000.

[31] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[32] Stéfan van der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[33] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[34] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[35] David P Doane. Aesthetic frequency classifications. *The American Statistician*, 30(4):181–183, 1976.

[36] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[37] Ping Zhong and Masao Fukushima. Regularized nonsmooth newton method for multi-class support vector machines. *Optimisation Methods and Software*, 22(1):225–236, 2007.

[38] Agapito Ledezma, Ricardo Aler, Araceli Sanchis, and Daniel Borrajo. Empirical evaluation of optimized stacking configurations. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on*, pages 49–55. IEEE, 2004.

[39] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 861–870. International Society for Optics and Photonics, 1993.

[40] Olvi L Mangasarian, W Nick Street, and William H Wolberg. Breast cancer diagnosis and prognosis via linear programming. *Operations Research*, 43(4):570–577, 1995.

[41] Qingping Tao. *Making efficient learning algorithms with exponentially many features*. PhD thesis, Citeseer, 2004.

[42] Brian Johnson and Zhixiao Xie. Classifying a high resolution image of an urban area using super-object information. *ISPRS journal of photogrammetry and remote sensing*, 83:40–49, 2013.

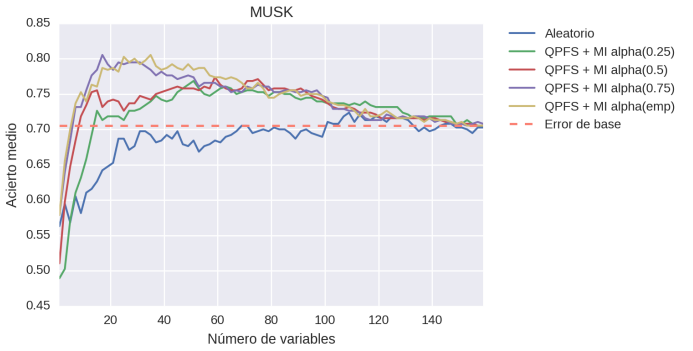
[43] E Ray Bareiss, Bruce W Porter, and Craig C Wier. Protos: An exemplar-based learning apprentice. In *Proceedings of the fourth international workshop on machine learning*, pages 12–23, 1987.

[44] Xiaoming Huo and Gábor J Székely. Fast computing for distance covariance. *Technometrics*, 58(4):435–447, 2016.

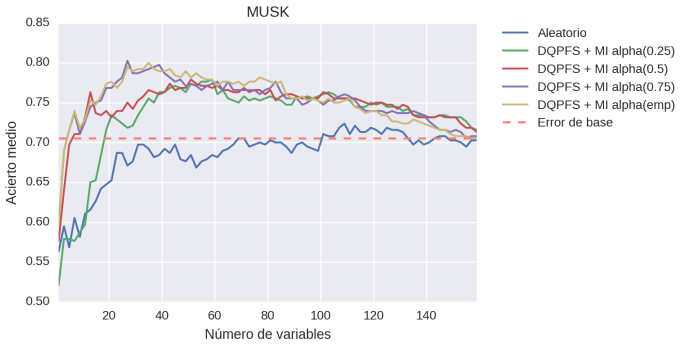
A

Promedio de aciertos

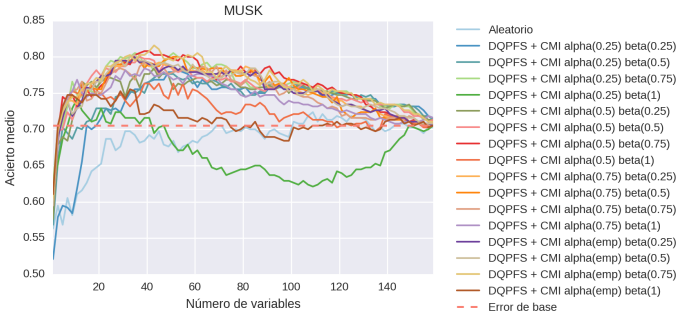
En este anexo se muestra el porcentaje promedio de acierto en clasificación usando Naive Bayes en función del número de variables y variando el valor de los parámetros para los conjuntos de datos mostrados en la Sección 4.3. Las pruebas se han realizado con $\alpha \in \{0.25, 0.5, 0.75, \hat{\hat{\alpha}}\}$, donde $\hat{\hat{\alpha}}$ representa el uso del α empírico detallado en la Sección 3.2, y $\beta \in \{0.25, 0.5, 0.75, 1\}$. En todas las figuras de este Anexo, la notación alpha(emp) representa el uso del parámetro $\hat{\hat{\alpha}}$.



(a) QPFS e Información Mutua.

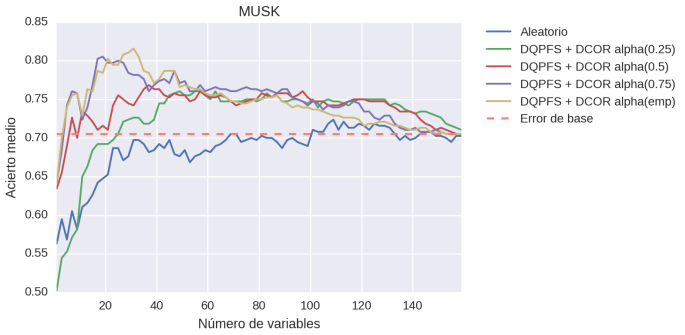


(b) DQPFS e Información Mutua.



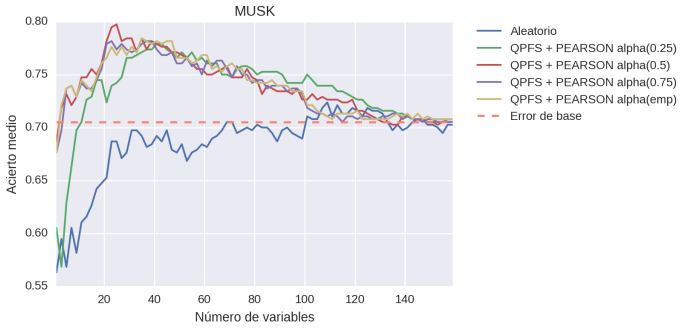
(c) DQPFS e Información Mutua Condicionada.

(d) DQPFS y Distancia de Covarianzas.



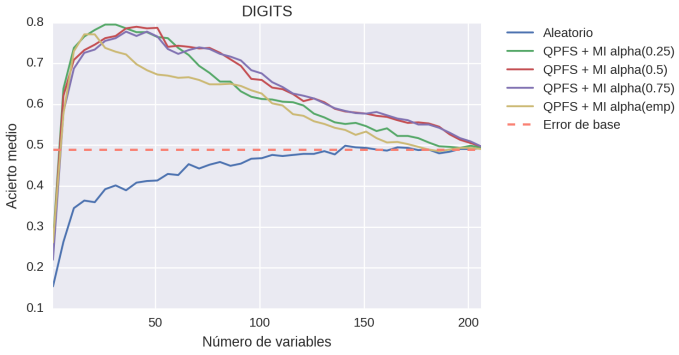
(e) DQPFS y Distancia de Correlaciones.

(f) DQPFS y Correlación de Pearson.

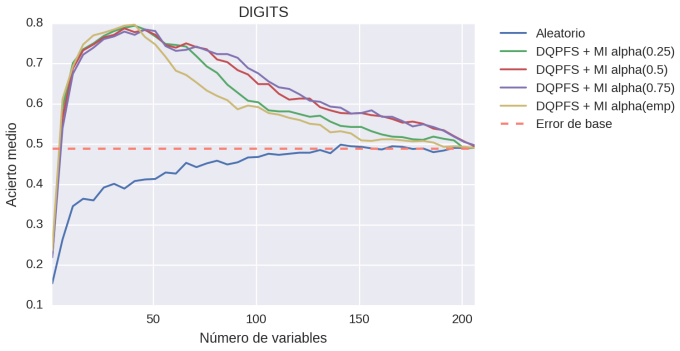


(g) QPFS y Correlación de Pearson.

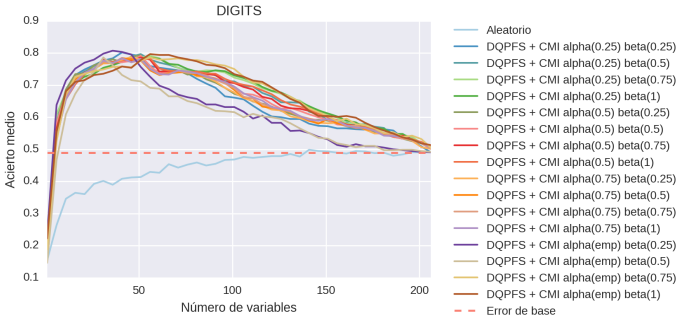
Figura A.1: Acierto en clasificación para el conjunto de datos MUSK para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud.



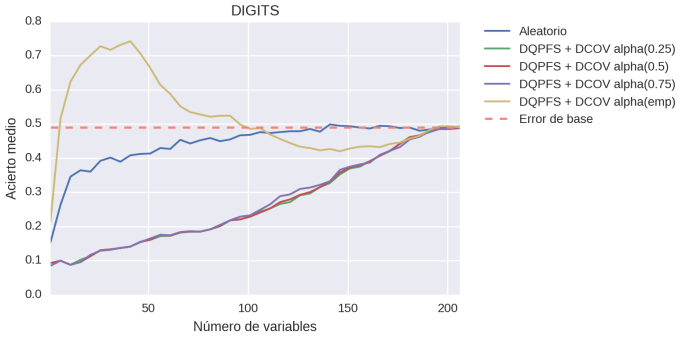
(a) QPFS e Información Mutua.



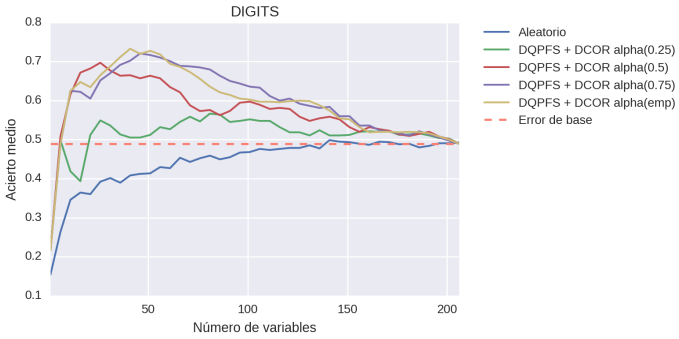
(b) DQPFS e Información Mutua.



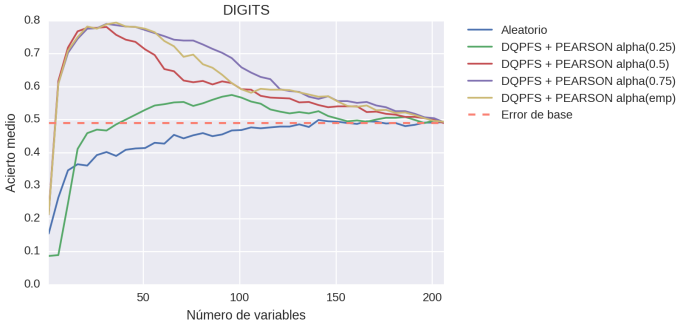
(c) DQPFS e Información Mutua Condicionada.



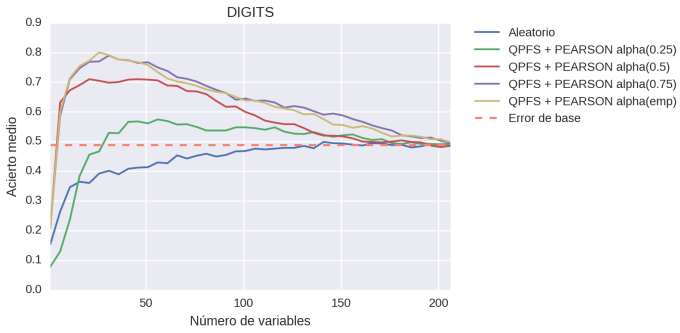
(d) DQPFS y Distancia de Covarianzas.



(e) DQPFS y Distancia de Correlaciones.

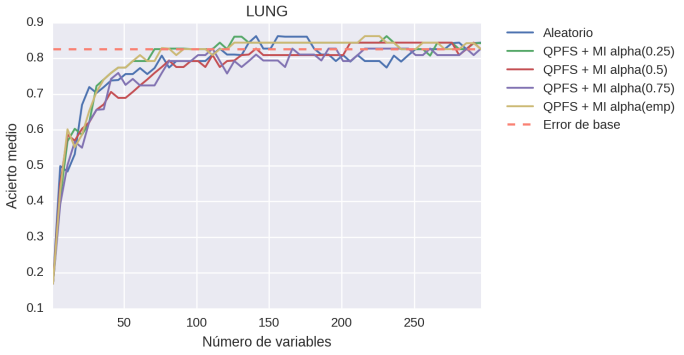


(f) DQPFS y Correlación de Pearson.

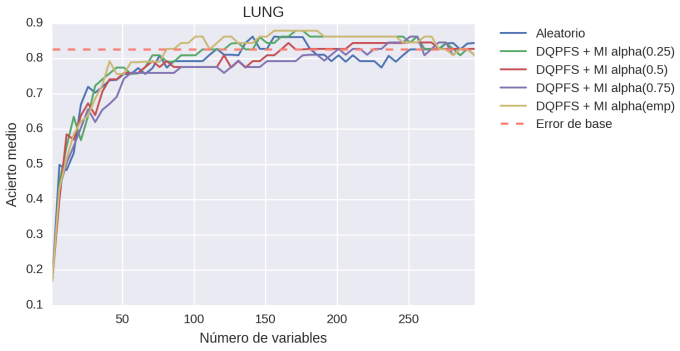


(g) QPFS y Correlación de Pearson.

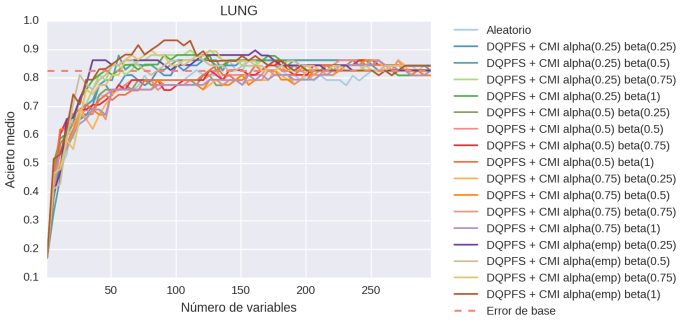
Figura A.2: Acierto en clasificación para el conjunto de datos DIGITS para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud.



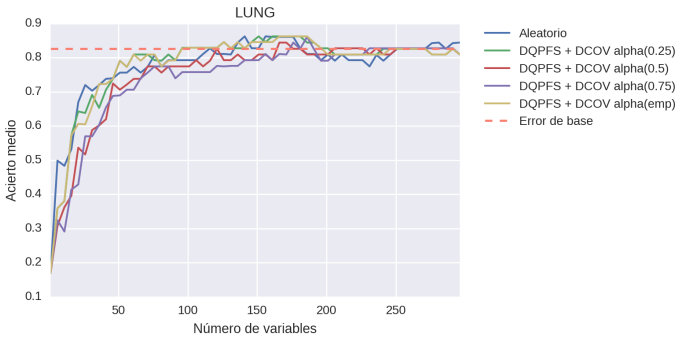
(a) QPFS e Información Mutua.



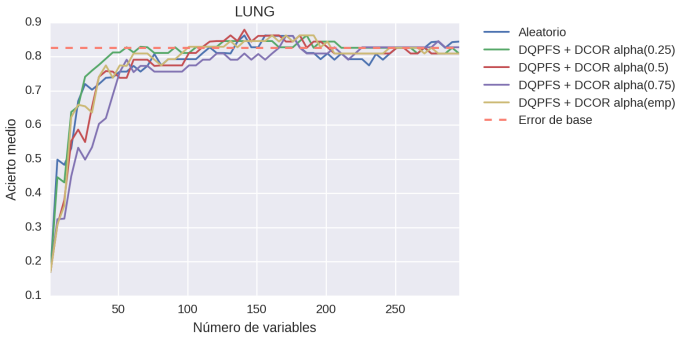
(b) DQPFS e Información Mutua.



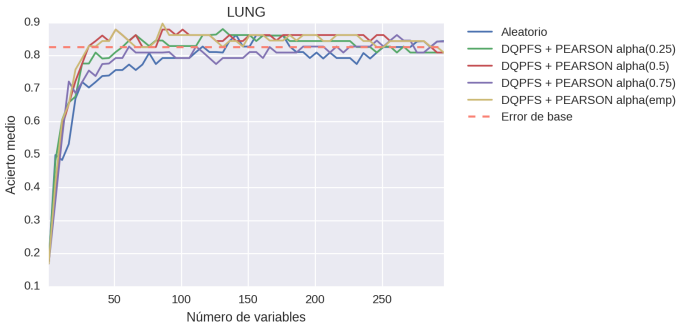
(c) DQPFS e Información Mutua Condicionada.



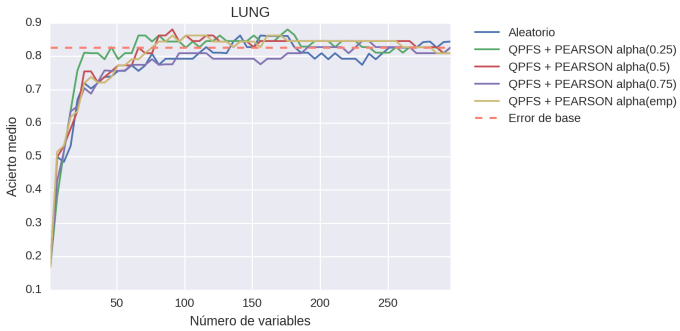
(d) DQPFS y Distancia de Covarianzas.



(e) DQPFS y Distancia de Correlaciones.

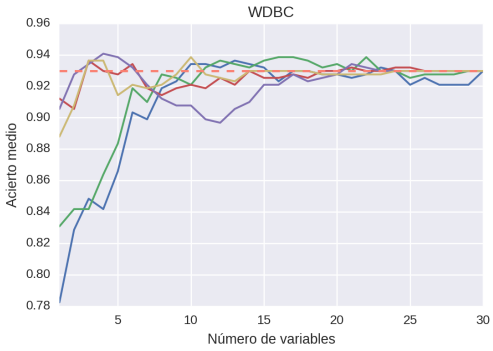


(f) DQPFS y Correlación de Pearson.

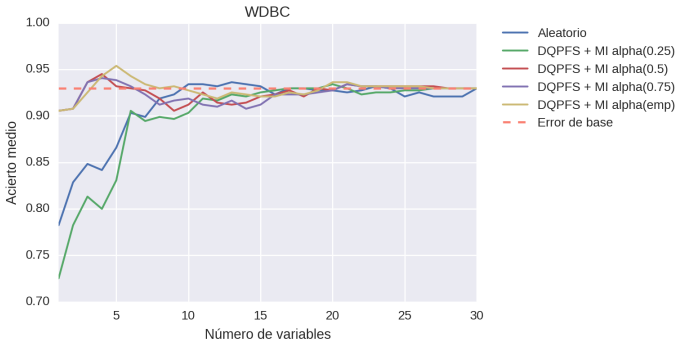


(g) QPFS y Correlación de Pearson.

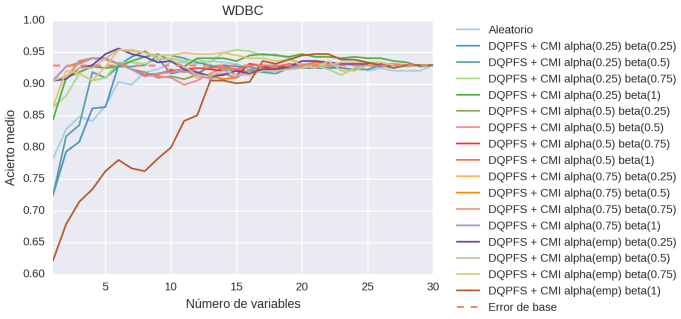
Figura A.3: Acierto en clasificación para el conjunto de datos LUNG para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud.



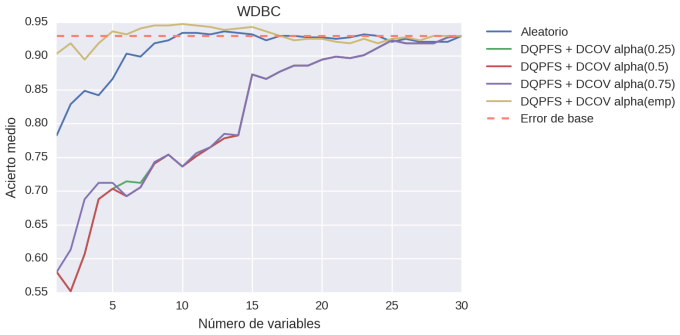
(a) QPFS e Información Mutua.



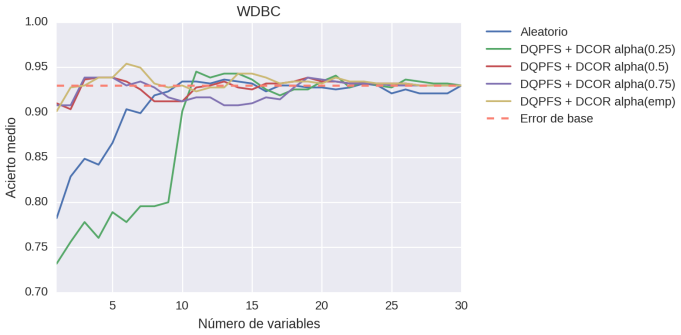
(b) DQPFS e Información Mutua.



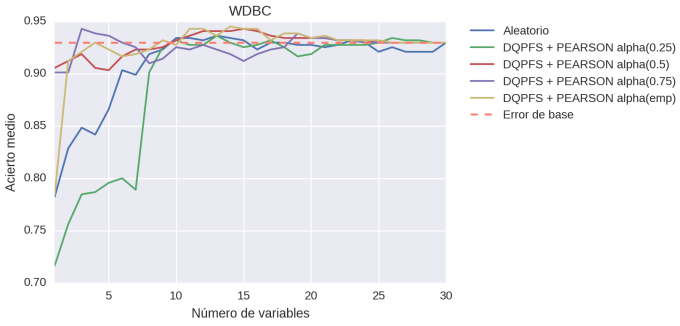
(c) DQPFS e Información Mutua Condicionada.



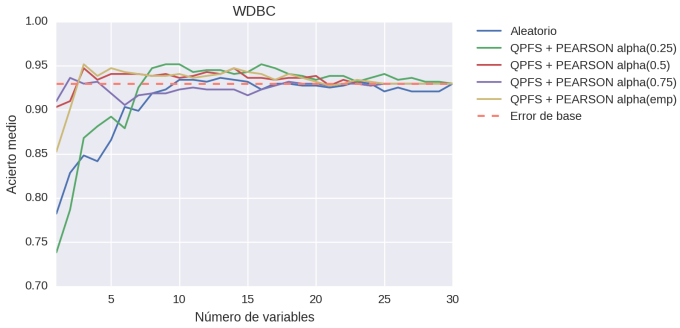
(d) DQPFS y Distancia de Covarianzas.



(e) DQPFS y Distancia de Correlaciones.

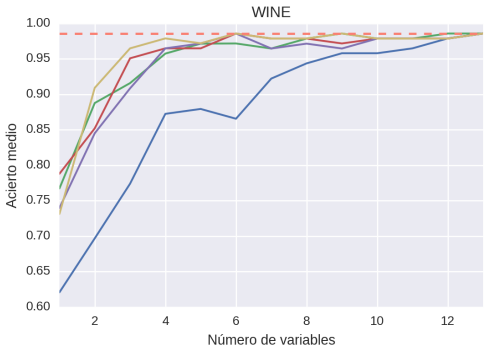


(f) DQPFS y Correlación de Pearson.

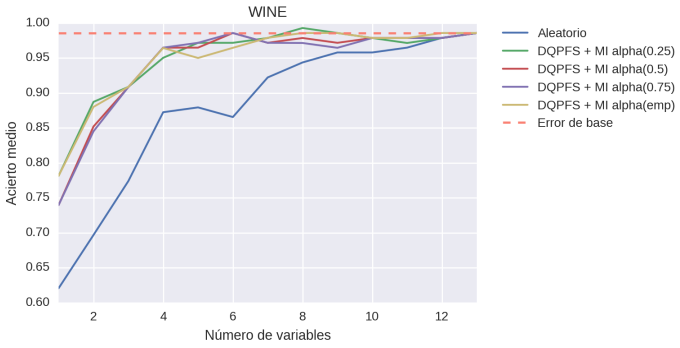


(g) QPFS y Correlación de Pearson.

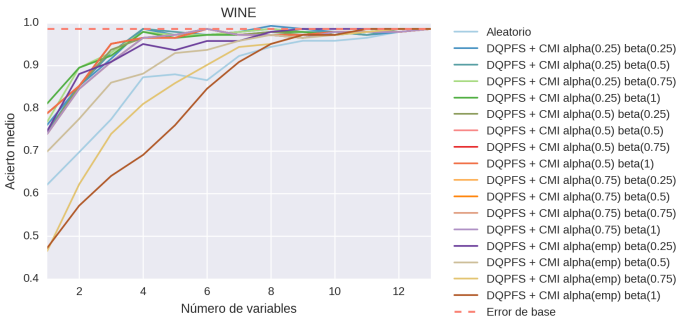
Figura A.4: Acierto en clasificación para el conjunto de datos WDBC para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud.



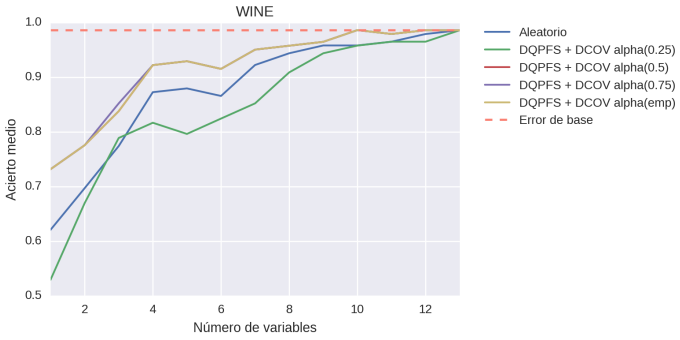
(a) QPFS e Información Mutua.



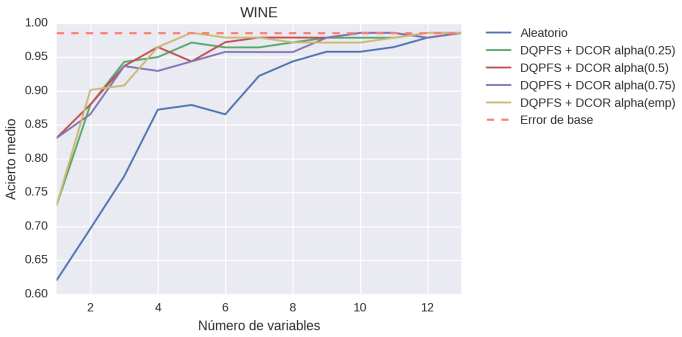
(b) DQPFS e Información Mutua.



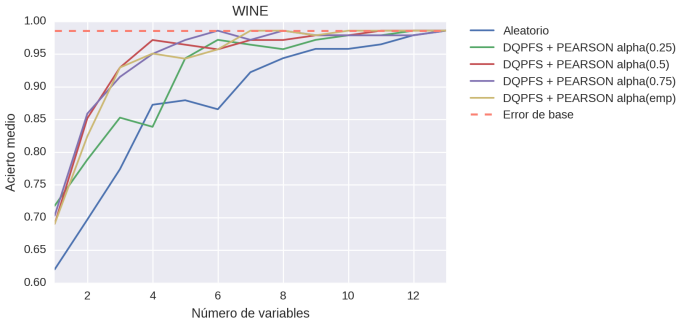
(c) DQPFS e Información Mutua Condicionada.



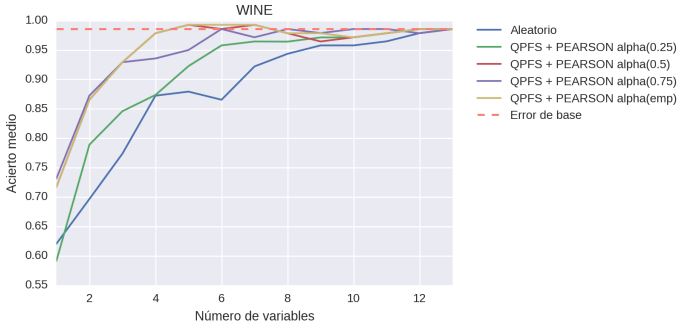
(d) DQPFS y Distancia de Covarianzas.



(e) DQPFS y Distancia de Correlaciones.

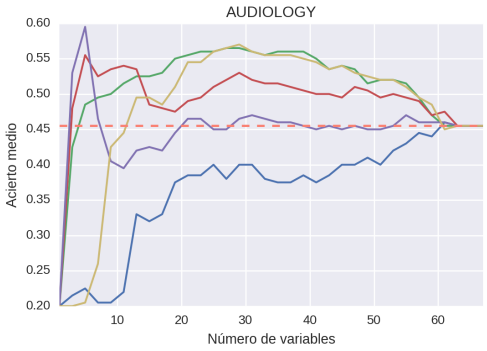


(f) DQPFS y Correlación de Pearson.

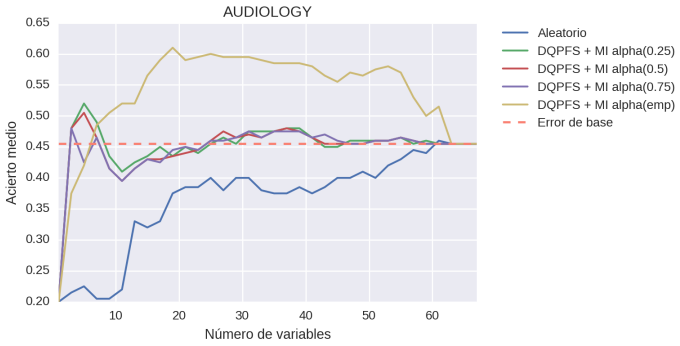


(g) QPFS y Correlación de Pearson.

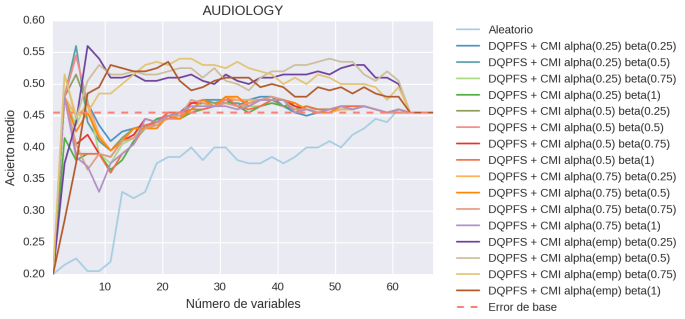
Figura A.5: Acierto en clasificación para el conjunto de datos WINE para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud.



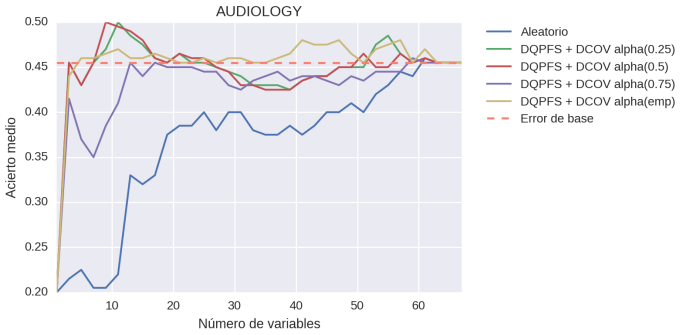
(a) QPFS e Información Mutua.



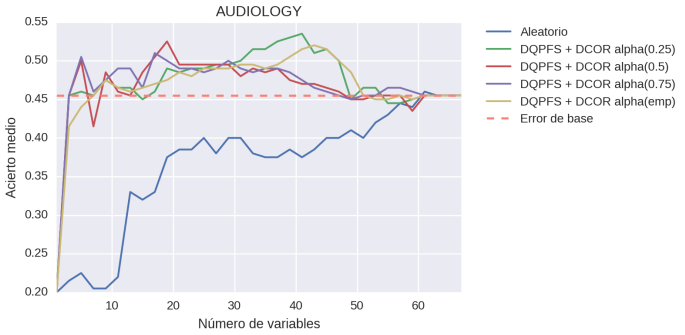
(b) DQPFS e Información Mutua.



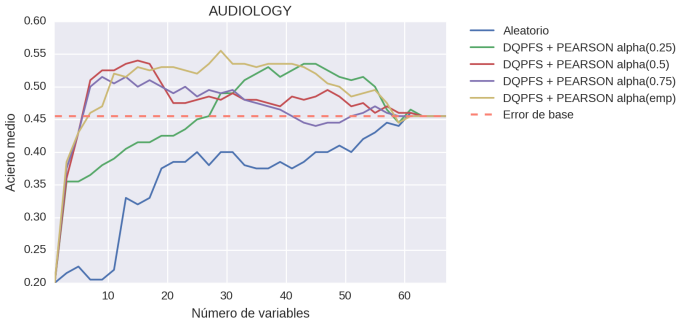
(c) DQPFS e Información Mutua Condicionada.



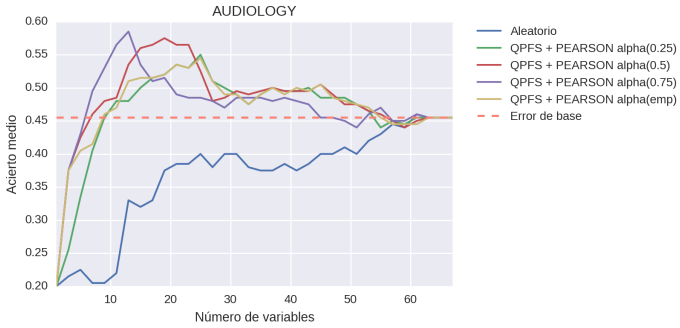
(d) DQPFS y Distancia de Covarianzas.



(e) DQPFS y Distancia de Correlaciones.

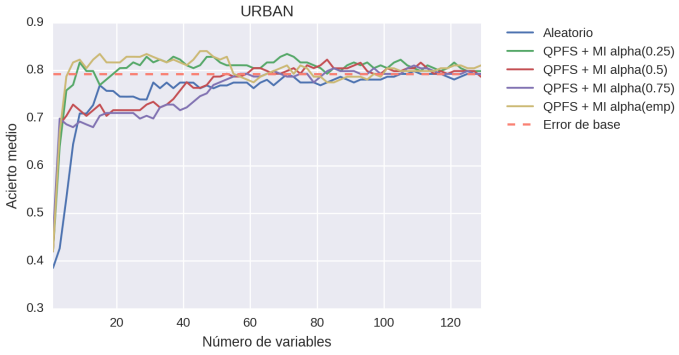


(f) DQPFS y Correlación de Pearson.

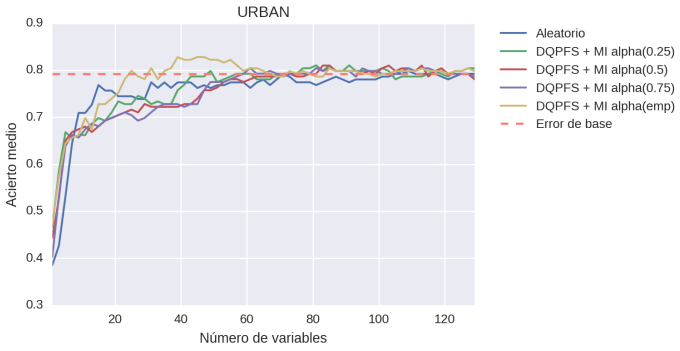


(g) QPFS y Correlación de Pearson.

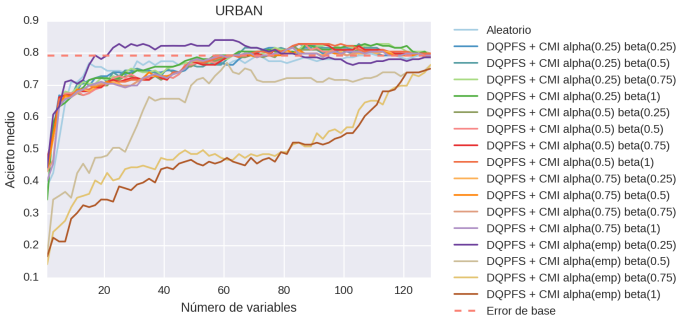
Figura A.6: Acierto en clasificación para el conjunto de datos AUDIOLOGY para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud.



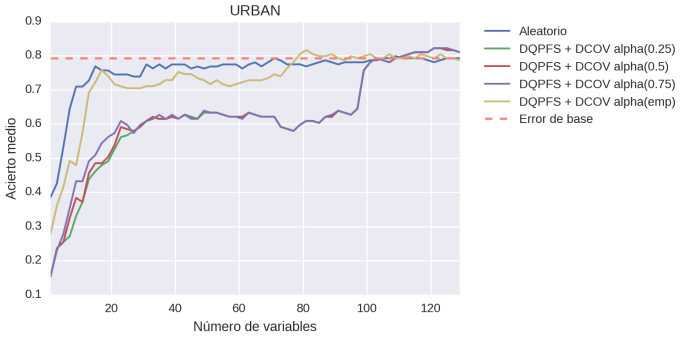
(a) QPFS e Información Mutua.



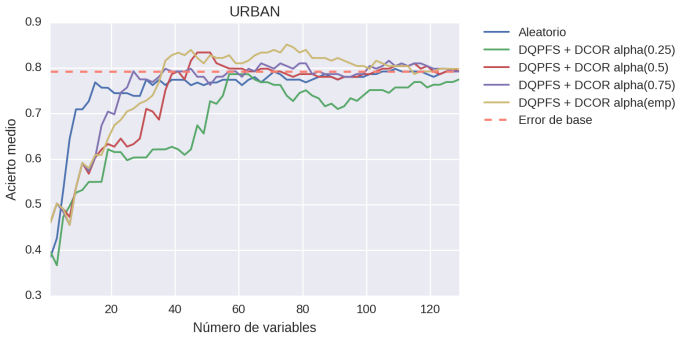
(b) DQPFS e Información Mutua.



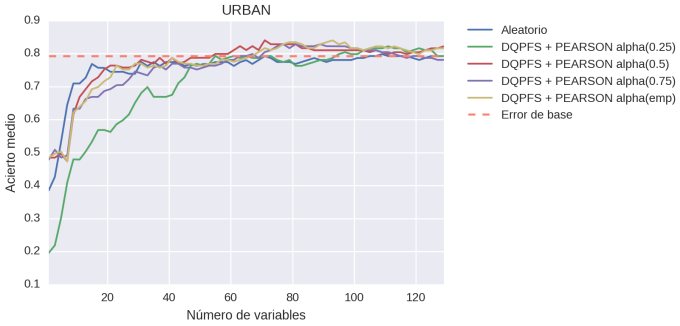
(c) DQPFS e Información Mutua Condicionada.



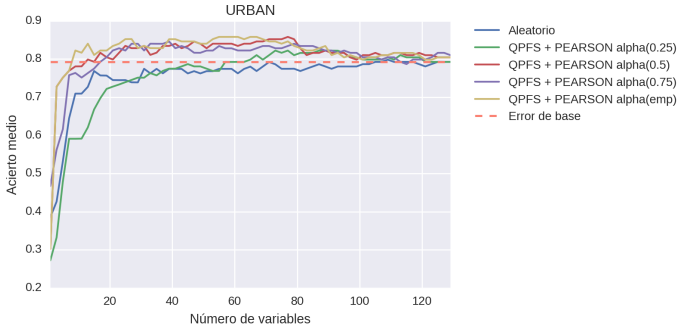
(d) DQPFS y Distancia de Covarianzas.



(e) DQPFS y Distancia de Correlaciones.



(f) DQPFS y Correlación de Pearson.



(g) QPFS y Correlación de Pearson.

Figura A.7: Acierto en clasificación para el conjunto de datos URBAN para los algoritmos de selección de variables QPFS y DQPFS y distintas medidas de similitud.